PROBABILISTIC SEMANTIC FRAMES

JIŘÍ MATERNA

Ph.D. Thesis

Natural Language Processing Centre
Faculty of Informatics
Masaryk University

April 2014

Jiří Materna: *Probabilistic Semantic Frames,* © April 2014

SUPERVISOR:
doc. PhDr. Karel Pala, CSc.

ABSTRACT

During recent decades, semantic frames, sometimes called semantic or thematic grids, have been becoming increasingly popular within the community of computational linguists. Semantic frames are conceptual structures capturing semantic roles valid for a set of lexical units, which can alternatively be seen as a formal representation of prototypical situations evoked by lexical units. Linguists use them for their ability to describe an interface between the syntax and the semantics. In practical natural language processing applications, they can be used, for instance, for the word sense disambiguation task, knowledge representation, or in order to resolve ambiguities in the syntactic analysis of natural languages.

Nowadays, lexicons of semantic frames are mainly created manually or semi-automatically by highly trained linguists. Manually created lexicons involve, for example, a well-known lexicon of semantic frames FrameNet, a lexicon of verb valencies known as VerbNet, or Czech valency lexicons Vallex and VerbaLex. These and other frame-based lexical resources have many promising applications, but suffer from several disadvantages. Most importantly, their creation requires manual work of trained linguists, which is very time-consuming and expensive. The coverage of the resources is then usually small or limited to specific domains.

In order to avoid the problems, this thesis proposes a method of creating semantic frames automatically. The basic idea is to generate a set of semantic frames and roles by maximizing the posterior probability of a probabilistic model on a syntactically parsed training corpus. A semantic role in the model is represented as a probability distribution over all its realizations in the corpus, and a semantic frame as a tuple of semantic roles, each of them connected with some grammatical relation. For every lexical unit from the corpus, a probability distribution over all semantic frames is generated. The probability of a frame corresponds to the relative frequency of its usage in the corpus for a given lexical unit. The method of constructing semantic roles is similar to inferring latent topics in Latent Dirichlet Allocation, thus, the probabilistic semantic frames proposed in this work are called LDA-Frames.

The most straightforward application is its use as a corpus-driven lexicon of semantic patterns that can be exploited by linguists for the language analysis, or by lexicographers for creating lexicons. The appropriateness of using LDA-Frames for these purposes is shown in the comparison with a manually created verb pattern lexicon from the Corpus Pattern Analysis project. The second application,

which demonstrates the power of LDA-Frames in this thesis, is the measurement of a semantic similarity between lexical units. It will be shown that the thesaurus built using the information from the probabilistic semantic frames overcomes a competitive approach based on the Sketch Engine.

This thesis is structured into two parts. The first part introduces basic concepts of lexical semantics and explores the state of the art in the field of semantic frames in computational linguistics. The second part is dedicated to describing all the necessary theoretical background, as well as the idea of LDA-Frames with its detailed description and evaluation. The basic model is provided along with several extensions, namely a non-parametric version and a parallelized computational model. The last sections of the thesis illustrate the usability of LDA-Frames in practical natural language processing applications.

PUBLICATIONS

Some ideas and figures from this thesis have previously appeared in the following publications:

- Jiří Materna. Parameter Estimation for LDA-Frames. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pages 482–486, Atlanta, Georgia, June 2013. Association for Computational Linguistics.
- Jiří Materna. LDA-Frames: An Unsupervised Approach to Generating Semantic Frames. In Alexander Gelbukh, editor, *Proceedings of the 13th International Conference CICLing 2012, Part I*, volume 7181 of *Lecture Notes in Computer Science*, pages 376–387. Springer Berlin / Heidelberg, 2012.
- Jiří Materna. Building a Thesaurus Using LDA-Frames. In Pavel Rychlý, Aleš Horák, editor, *Recent Advances in Slavonic Natural Language Processing*, pages 97–103. Tribun EU, 2012.
- Jiří Materna and Karel Pala. Using Ontologies for Semi-automatic Linking VerbaLex with FrameNet. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta Malta, 2010. (Contribution: 80 %)
- Jiří Materna. Linking VerbaLex vith FrameNet: Case Study for the Indicate Verb Class. In Petr Sojka, Aleš Horák, editor, *Recent Advances in Slavonic Natural Language Processing*, pages 89–94. Tribun EU, 2010.
- Jiří Materna. Linking Czech Verb Valency Lexicon VerbaLex with FrameNet. In *Proceedings of 4th Language and Technology Conference*, pages 215–219, Poznań, 2009
- Jiří Materna. Czech Verbs in FrameNet Semantics. In *Czech in Formal Grammar*, pages 131–139, Lincom, München, 2009.

Other author's publications include:

- Jiří Materna. Determination of the topic consistency of documents. In Radim Jiroušek, Jiří Jelínek, editor, *Znalosti 2011*, pages 148–158. Fakulta elektrotechniky a informatiky, VŠB – Technická univerzita Ostrava, 2011.
- Jiří Materna and Juraj Hreško. A Bayesian Approach to Query Language Identification. In Pavel Rychlý Aleš Horák, editor, *Recent Advances in Slavonic Natural Language Processing*, pages 111–116. Tribun EU, 2011. (Contribution: 80 %)

- Jiří Materna. Domain Collocation Identification. In *Recent Advances in Slavonic Natural Language Processing*, Faculty of Informatics, Masaryk University, Brno, 2009.
- Lubomír Popelínský and Jiří Materna. On key words and key patterns in Shakespeare's plays. In *Znalosti 2009*, Nakladatelstvo STU, Bratislava, 2009. (Contribution 20 %)
- Jiří Materna. Automatic Web Page Classification. In *Recent Advances in Slavonic Natural Language Processing*, Faculty of Informatics, Masaryk University, Brno, 2008.
- Lubomír Popelínský and Jiří Materna. Keyness and Relational Learning. In *International conference Keyness in Text*, University of Siena, Siena, 2007. (Contribution 30 %)
- Jiří Materna. Keyness in Shakespeare's Plays. In *Recent Advances in Slavonic Natural Language Processing*, Faculty of Informatics, Masaryk University, Brno, 2007.

ACKNOWLEDGMENTS

First of all I would like to thank my supervisor Karel Pala for his support and encouragement on one hand, and his understanding for my duties at the Seznam.cz company on the other hand. Many thanks go to my employer Seznam.cz and all my colleagues for an opportunity to be a part of a mutually enriching environment, where research and applications together result in amazing technological products. I would also like to thank my colleagues from the NLP lab for valuable discussions of the issues of probabilistic semantic frames during the last years. I am very grateful to Tomáš Brázdil and Vladimír Kadlec who provided me with numerous remarks on the draft of this thesis, Steven Schwartz for his comments regarding my English, and André Miede for his nice LaTeXtemplate. Finally, I would like to thank my parents and my partner Eliška Mikmeková for their love and support.

CONTENTS

I SEMANTIC FRAMES 1

1 INTRODUCTION 3

1.1 Lexical Semantics 4

1.1.1 Homonymy and Polysemy 5

1.2 Semantic Relations 6

1.2.1 Synonymy 6

1.2.2 Hyponymy/hypernymy 7

1.2.3 Meronymy/holonymy 7

1.2.4 Antonymy 7

1.3 Ontologies and Frames 7

1.3.1 Ontologies 8

1.3.2 WordNet 8

1.3.3 Frames 9

1.4 Probabilistic Frames 10

1.4.1 Practical Semantic Frames 11

1.4.2 Related Work 15

1.5 Structure of the Thesis 16

2 SEMANTIC FRAMES FOR NATURAL LANGUAGE PROCESSING 17

2.1 Semantic Roles 18

2.2 Connection with Syntax 21

2.3 Verb Valencies and Semantic Frames 22

2.3.1 Obligatory vs. Facultative Valencies 22

2.3.2 Verb Classes 23

2.3.3 Verb-specific vs. General Semantic Roles . . . 24

2.3.4 Relations Between Verb Valencies and Semantic Frames 24

2.4 Motivation for Identifying Semantic Frames 26

3 EXISTING FRAME-BASED LEXICAL RESOURCES 29

3.1 PropBank 29

3.2 VerbNet 30

3.3 The Berkeley FrameNet 30

3.3.1 Semantic Frames 31

3.3.2 Frame Elements 32

3.3.3 FrameNet relations 33

3.3.4 Semantic types 34

3.4 FrameNets in other languages 34

3.4.1 SALSA 34

3.4.2 Spanish FrameNet 35

3.5 BRIEF 36

3.6 Vallex 36

3.7 VerbaLex 37

II PROBABILISTIC SEMANTIC FRAMES 39

4 PROBABILISTIC GRAPHICAL MODELS 41
4.1 Bayesian Networks 41
4.1.1 Model Representation 43
4.2 Generative vs. Discriminative models 45
4.3 Statistical Inference in Bayesian Networks 48
4.3.1 Variational Inference 49
4.3.2 Sampling 50

5 SELECTED PROBABILITY DISTRIBUTIONS 59
5.1 The Binomial and the Multinomial distributions . . . 60
5.2 The Beta and the Dirichlet distributions 61
5.3 The Gamma Distribution 64

6 TOPIC MODELS 67
6.1 Latent Semantic Analysis 68
6.2 Probabilistic Latent Semantic Analysis 71
6.3 Latent Dirichlet Allocation 74

7 LDA-FRAMES 77
7.1 Model Description 79
7.2 The Gibbs Sampler for LDA-Frames 82
7.2.1 Sampling Frames 85
7.2.2 Sampling Roles 87

8 NON-PARAMETRIC LDA-FRAMES 91
8.1 Dirichlet Process 91
8.2 Chinese restaurant process and Stick breaking process 93
8.3 Estimating the number of roles 96
8.4 Estimating the Number of Frames 98
8.4.1 Hierarchical Dirichlet Process 98
8.4.2 Fully Non-Parametric Model 102
8.4.3 Sampling Unbounded Number of Frames . . 103
8.5 Gibbs Sampler for Non-Parametric LDA-Frames . . . 105

9 HYPERPARAMETER ESTIMATION 109
9.1 Hyperparameter Estimation Method for Parametric LDA-Frames 110
9.2 Hyperparameter Estimation Method for Non-Parametric LDA-Frames 112

10 DISTRIBUTED INFERENCE ALGORITHMS 117
10.1 Parallel LDA-Frames 117
10.2 Parallel Non-parametric LDA-Frames 120

11 EVALUATION 123
11.1 Data Preparation 123
11.2 Implementation 128
11.3 Perplexity 131
11.3.1 Convergence 131
11.4 Reconstruction Abilities 134

11.5 Scalability 135
11.6 Comparison with CPA 137
11.6.1 Case Study: verbs 'access', 'submit' and 'pour' 139
11.6.2 Comparison on 300 most frequent lexical units 144
11.7 Semantic Similarity of Lexical Units 145
12 CONCLUSION 149

III APPENDIX 151
A COMPILATION AND USAGE INSTRUCTIONS 153
A.1 Introduction 153
A.2 Dependencies 153
A.3 Building binaries from the source codes on Ubuntu 13.10. 153
A.4 Example of the Usage 154
B LDA-FRAMES MODEL – SELECTED FRAMES 155

BIBLIOGRAPHY 157

INDEX 169

LIST OF FIGURES

Figure 1.1 Semantic frames evoked by lexical units *consume* and *eat*. 10
Figure 2.1 Example of the Prepositional Phrase Attachment problem. 18
Figure 3.1 PropBank example entry for lexical unit `'open'`. 30
Figure 3.2 FrameNet frame example: `Ingestion`. 32
Figure 3.3 FrameNet relation example: `Crime_scenario`. . 33
Figure 3.4 SALSA annotation example: `Categorization`. . 35
Figure 3.5 Example of the Vallex frame for 'vařit'. 36
Figure 3.6 Example of the VerbaLex frame for 'vařit'. . . . 38
Figure 4.1 Bayesian Network. 42
Figure 4.2 Plot of the univariate Gaussian showing the mean μ and the standard deviation σ. 43
Figure 4.3 Probabilistic graphical model for the distribution of people's heights in the population. . . . 44
Figure 4.4 Generative and discriminative types of classifiers. 47
Figure 4.5 Logistic function. 48
Figure 5.1 The binomial distribution with $p = 0.6$ and $n = 10$. 60
Figure 5.2 The beta distribution for various values of the parameters. 62
Figure 5.3 Examples of three-dimensional Dirichlet distributions. 63
Figure 5.4 Probability density functions for various parameters of the gamma distribution. 64
Figure 6.1 Generative process for probabilistic topic models. 71
Figure 6.2 Graphical model for PLSA. 72
Figure 6.3 Graphical model of latent Dirichlet allocation. 75
Figure 7.1 Graphical model for LDA-frames. 81
Figure 8.1 Chinese restaurant process. 94
Figure 8.2 Estimating the number of roles in LDA-frames. 96
Figure 8.3 Hierarchical Dirichlet process. 99
Figure 8.4 Chinese restaurant franchise. 101
Figure 8.5 Graphical model of Non-parametric LDA-frames. 102
Figure 11.1 Example of the typed dependency parse for the sentence "I saw the man who loves you". . 124
Figure 11.2 Directory tree structure. 129
Figure 11.3 A screenshot of the semantic frame representation on the LDA-Frames web site – two most likely frames for lexical unit 'eat'. 130
Figure 11.4 Convergence of the LDA-Frames algorithm. . . 132

Figure 11.5 Convergence and the number of components in the Non-Parametric LDA-Frames algorithm. 133
Figure 11.6 Perplexities for different values of F and R on the *synthetic* data set. 134
Figure 11.7 Convergence of the test perplexity for the parallelized LDA-Frames algorithm. 136
Figure 11.8 Speedup for 1 to 10 cores using the parallelized LDA-Frames algorithm. 136
Figure 11.9 Screenshot of the PDEV entry for verb 'access'. 138
Figure 11.10 Average number of words in the similarity sets. 146
Figure 11.11 Comparison of the Sketch Engine thesauri with the LDA-frames thesauri. 147

Table 6.1 The bag-of-words representation of text documents. 69
Table 6.2 Reduced 2-dimensional representation of the documents. 71
Table 7.1 Example of training data. 77
Table 7.2 Extended example of a training data set. 78
Table 7.3 Semantic role labels 78
Table 11.1 Semantic roles along with their lexical realizations. 126
Table 11.2 Lexical units and their semantic frames. 127
Table 11.3 Evaluation data sets. 128
Table 11.4 Matching between Corpus Pattern Analysis (CPA) and LDA-Frames. 142
Table 11.5 Statistics of the matching between CPA and LDA-Frames. 142
Table 11.6 Quality of the automatic assignment of LDA-Frames. 143
Table 11.7 The correspondence of LDA-Frames and CPA patterns. 145
Table B.1 Top-ten realizations of semantic roles in LDA-Frames. 156

LIST OF ALGORITHMS

Algorithm 4.1 Rejection sampling algorithm for N iterations. 52
Algorithm 4.2 Metropolis sampling algorithm for N iterations. 55
Algorithm 4.3 Gibbs sampling algorithm for N iterations. . . 57

Algorithm 6.1 Generative process for PLSA. 72
Algorithm 6.2 Generative process for LDA. 75

Algorithm 7.1 Generative process for LDA-Frames. 80
Algorithm 7.2 Sampling from the categorical distribution over frames. 88

Algorithm 8.1 Algorithm for sampling a new frame. 104
Algorithm 8.2 Algorithm for sampling non-parametric LDA-Frames. 106

Algorithm 9.1 Algorithm for sampling the γ hyperparameter. 114
Algorithm 9.2 Algorithm for sampling the α hyperparameter. 115

Algorithm 10.1 Algorithm for distributed LDA-Frames. 119

ACRONYMS

BNC British National Corpus

CPA Corpus Pattern Analysis

CPU Central Processing Unit

CVF Complex Valency Frame

EM Expectation-Maximization

FE Frame Element

IR Information Retrieval

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

LSI Latent Semantic Indexing

MCMC Markov Chain Monte Carlo

MLE Maximum-Likelihood Estimation

NLP Natural Language Processing

PCFG Probabilistic Context-Free Grammar

PDEV Pattern Dictionary of English Verbs

PGM Probabilistic Graphical Model

PLSA Probabilistic Latent Semantic Analysis

PWN Princeton WordNet

SR Semantic Role

SVD Singular Value Decomposition

VPS Verb Pattern Sample

WSD Word Sense Disambiguation

WSI Word Sense Induction

Part I

SEMANTIC FRAMES

You shall know a word by the company it keeps.

(*Firth, 1957*)

1 INTRODUCTION

Word sense ambiguity is a well-known phenomenon typical for all natural languages throughout the world, which must be solved by speakers when communicating. The ability to differentiate word meanings and to choose the correct meaning based on a discourse or gesture context is an intrinsic capability of humans. It has been shown (Piantadosi et al., 2011) that the word sense ambiguity is an essential part of all languages, allowing greater communicative efficiency, which cannot be avoided even in artificial languages such as Esperanto (Zamenhof, 1991) if they were used massively. In fact, language ambiguity allows communication systems to be short, simple and efficient.

In the computational Natural Language Processing (NLP), by contrast, an ambiguity at all levels including word sense ambiguity is a source of many problems. Many natural language processing applications like machine translation and information retrieval require techniques enabling, at least partially, unambiguous semantic parsing and understanding.

The task of identifying the correct sense of a word based on the textual context is usually called **word sense disambiguation**. It assumes that the set of possible word senses is known and that it is required to select the most appropriate sense. The set of word senses, however, is usually defined manually, which is a time-consuming task for humans that typically leads to a subjectively biased selection. One possible way to avoid these problems is to find word senses automatically based on text corpora. The task of identifying the word senses of a word automatically is called **word sense induction**.

word sense disambiguation

word sense induction

An important issue of dealing with multiple word senses in computers is their formal representation. Conventional dictionaries for humans, which are essentially lists of lexical entries introduced and alphabetized by a head-word, represent different senses of the head-word by sense numbers accompanied with a definition described in a natural language. Such form of representation is convenient for people. As it comes to representing word senses in computers, however, we need something more formal (the definitions themselves, written in a natural language, can be ambiguous and require the semantic parsing).

Nowadays, there is a trend to build large domain independent electronic lexical databases containing as much semantic information as possible. Probably the best known semantic database is the Princeton WordNet (Fellbaum, 1998). It is a large lexical resource

of American English, developed under the direction of George A. Miller, where nouns, verbs, adjectives and adverbs are grouped into sets of synonyms (**synsets**). The synsets are interlinked by means of conceptual-semantic and lexical relations (e.g. synonymy, antonymy, hypero/hyponymy, meronymy and holonymy).

synset

Another complex electronic lexical resource for English is known as FrameNet (Ruppenhofer et al., 2006), which is based on the Frame Semantics proposed by Charles J. Fillmore (Fillmore, 1982). In Frame Semantics, a word meaning is described in a relation to the semantic frame, which consists of a targeted lexical unit (pairing of a word with a sense), frame elements (its semantic dependants) and relations between them.

WordNet and FrameNet are examples of lexical resources that can be exploited for the formal representation of word senses, e.g. (Li et al., 1995; Banerjee and Pedersen, 2002; Kolte and Bhirud, 2008; Siemiński, 2011), but suffer from some disadvantages stemmed from the fact that they are mostly created manually or semi-automatically. These drawbacks along with the proposed solution, which is the main topic of this thesis, will be described later. Nevertheless, before we start to explore the electronic lexical resources deeply, it is inevitable to define the most important terms related to linguistic semantics.

1.1 LEXICAL SEMANTICS

Linguistic semantics is generally defined as the study of meaning in languages. We should ask, however, what is the meaning of 'meaning' in terms of lexical semantics. First of all, it is necessary to distinguish between use and mention, which is the concept known from analytic philosophy (Borchert, 2005). For example, we can use the word 'semantics' in two sentences with completely different aspects:

(1.1) *Semantics is an important part of the natural language processing.*

(1.2) *The word 'semantics' has nine letters.*

In sentence (1.1), the word 'semantics' is interpreted normally as a reference to a field of natural language processing and is implied that it is used. On the other hand, in sentence (1.2), the word 'semantics' is interpreted as a chain of letters and is said to be mentioned. Following the notation from this example, the mentioned words will be surrounded by the quotation marks for the rest of this thesis.

When speaking about words, it is worth describing the difference between **types** and **tokens**, and between **words** and **word-forms** (Lyons, 1977a). The distinction between type and token has been introduced into semantics by an American philosopher Peirce (1931 – 1958). Types are generally abstract and unique sort of things, in

type
token
word
word-form

contrary to tokens, that are their particular instances. Let us describe the relationship by considering the following example

(1.3) *to be or not to be*

This famous opening phrase of a monologue from William Shakespeare's play Hamlet consists of 4 types (unique words) and 6 tokens. Briefly said, tokens are instantiations of their types.

The distinction between words and word-forms is for many people unclear and the term 'word' is usually used with two different meanings. To be accurate, we say that word-forms are different grammatical forms of the same word. For example, 'mouse' and 'mice' are different word-forms of the word 'mouse'.

Having prepared the necessary terminology, we can proceed to the description of lexical semantics. **Lexical semantics** is a subfield of linguistic semantics dealing with the meaning of individual words. The units of meaning in lexical semantics are called **lexical units**. Unlike **compositional semantics**, which is concerned with meanings of whole sentences, we can store all lexical units in a mental **lexicon**. No such mental lexicon is ever complete because we are continually learning new words and forgetting those that are not used very often. On the other hand, according to Chomsky (1969a), when producing a sentence, we are creating new constructions that have never been used by a recursive application of a fixed set of grammatical and semantic rules.

lexical semantics

lexical unit

compositional semantics

lexicon

In order to be able to use lexical units in sentences correctly, we need to accompany them with some information about their prototypical behavior. One approach to store such information in electronic lexical databases is by using a structure called the **semantic frame**.

semantic frame

In this work, we will focus particularly on what individual lexical units mean, why they mean what they mean and how we can represent their meaning. The specific representation of lexical meaning will be discussed in detail.

1.1.1 *Homonymy and Polysemy*

In the field of lexical semantics, linguists distinguish two types of ambiguity – **homonymy** and **polysemy**. The most important difference between homonymy and polysemy is defined in terms of a word etymology.

homonymy

polysemy

Let us begin to differentiate them by looking at an example. Linguists say that 'bank_1' (bank of a river) and 'bank_2' (financial institution) are distinct but homonymous words, but that 'mouth' is a simple polysemous word with several different senses like "body organ" or "entrance of a cave". The criterion is based on the

knowledge of the historical derivation of words. Words 'bank$_1$' and 'bank$_2$' were derived from distinct words in an earlier stage of the language, but 'mouth' in all meanings came from one word (Lyons, 1977b).

In practice, the etymological criterion is not always distinctive and the situation is much more complicated. For the purposes of this work, however, we do not need to distinguish between homonymy and polysemy. Thus, we will use both terms interchangeably in further text.

1.2 SEMANTIC RELATIONS

Semantic relations can simply be defined as relations between meanings or concepts. For instance, the concept *car* can be expressed by different words with the same or similar meaning like 'car', 'automobile', 'auto', etc. This kind of semantic relation is called a synonymy.

The set of all semantic relations may not be finite. According to Bean and Green (2001), the inventory of relations contains both a closed set of relationships, including mainly equivalence and hierarchical relationships, and an open set of relations (especially between verbs). Every time a new verb is introduced, a new type of relationship may arise. The verb 'love', for example, defines an asymmetric relationship between two entities, where one entity (typically a person) is in love with the other.

Since an exact and deep exploration of semantic relations and their classification (which is still an open problem) is not necessary for understanding semantic frames, we will only limit this to the most important and general semantic relations.

1.2.1 *Synonymy*

synonymy

Synonymy is one of the most important relationships between lexical units. According to the definition usually attributed to Leibniz, synonyms are different words, which can replace one another in any context without changing the meaning of the proposition. This definition, however, is too strict and is fulfilled only by a small number of very similar words. That is why the **near synonymy** is used instead of the Leibniz's synonymy. With accordance to the definition of the near synonymy, two different words are synonyms if one can be replaced by the other in the same context without changing the sense of the sentence. An example of such synonymy relation are the words 'eat' and 'consume'.

near synonymy

1.2.2 *Hyponymy/hypernymy*

Hyponymy and **hypernymy** are relationships between two lexical units, where one's semantic range is within/outside of the other's. In computer science, this type of relationship is sometimes called IS-A, TYPE-OF, or lexical inheritance relation because all hyponyms must inherit all properties of their hypernyms. For example, the word 'animal' is a hypernym of 'dog' and 'car' is a hyponym of 'vehicle'.

hyponymy
hypernymy

1.2.3 *Meronymy/holonymy*

Meronymy and **holonymy** are relationships between a whole and its parts. For example, 'wheel' is a meronym of 'car' and 'house' is a holonym of 'door'. This relationship is, similarly to the hyponymy/hypernymy relation, transitive.

meronymy
holonymy

1.2.4 *Antonymy*

Lexical units with opposite meaning are interlinked by the **antonymy** relationship. Words may have several different antonyms, depending on their meaning. For instance, both 'long' and 'tall' can be antonyms of 'short'. The definition of antonymy may be a little bit confusing. For example, antonym is not the same as negation: telling that someone is not rich does not mean that he or she is poor although these two words are antonyms.

antonymy

1.3 ONTOLOGIES AND FRAMES

Assuming that the independent meanings of words are identified, we need to know how to connect the semantic information from the lexicon with other words and with the background knowledge to determine the meaning of whole clauses or larger text sequences.

It turns out that it is not easy to isolate the meaning of a word from its context and the outer world. When putting pieces of word senses together, it is necessary to investigate the role of contextual information and to combine a linguistic knowledge with a background knowledge of the world and the word's prototypical behavior. People, when listening to a speaker, have a very active role. Using what has been said together with known language patterns, they make inferences about what the speaker meant (Saeed, 2009).

For many years, there has been an attempt to formalize the information about contextual and world knowledge in electronic databases, which has led to the development of electronic ontologies and networks of frame structures such as semantic frames.

1.3.1 *Ontologies*

The term ontology has its origin in philosophy and represents a philosophical study of the nature of being as well as the basic categories of being and their relations. It was Aristotle who first constructed a well-defined ontology. In his Metaphysic (Sachs, 1999) he analyzed the simplest elements for which the mind reduces the real world of reality.

Moving back to computer science, ontology is a formal representation of a set of concepts[1] and relationships between them. In order to explain how the ontologies are constructed and what are they good for, we can refer to logical systems. The existential quantifier in logic is a notation for asserting that something exists, but logic itself has no vocabulary for describing things that exist. Ontology fills that gap. It is the study of the existence of all kinds of entities, abstract and concrete, that make up the world or a particular domain.

domain-specific ontology

upper ontology

Domain ontologies, sometimes called **domain-specific ontologies**, model a specific part of the world. Universal, non-domain ontologies, are known as **upper ontologies**. An upper ontology is a model of the common world, which describes general concepts that are the same across all domains. There are several upper ontologies including commercial, e.g. Cyc (Lenat, 1995), as well as freely available ones, e.g. SUMO (Niles and Pease, 2001) or Dolce (Gangemi et al., 2002).

1.3.2 *WordNet*

One of the most famous electronic lexical resources combining linguistics with the idea of ontologies is WordNet. The Princeton WordNet (PWN) (Fellbaum, 1998) is a large lexical database of the English language, developed under the direction of George A. Miller at Princeton University. Its design has been inspired by psycholinguistic research and computational theories of human lexical memory (Miller et al., 1993). Entries in PWN are made of nouns, verbs, adjectives and adverbs grouped into sets of cognitive synonyms called synsets, each expressing a distinct concept. Different senses of polysemous words (literals) are distinguished by their sense IDs (positive integers). Synsets in PWN are interlinked by a means of semantic relations, which form the net, wherein the most important relations include synonymy, hyponymy/hypernymy, meronymy/holonymy and antonymy.

Whilst at the beginning of the project, WordNet was intended to be a psycholinguistic study of the human mind, nowadays it is used in many branches of natural language processing as a general purpose reference source for the English language. It is freely available on

1 When speaking about concepts we mean rather their labels because as such they are not linguistic entities.

the web[2], and furthermore, it can be used, copied, modified and distributed as a part of commercial applications.

Since the Princeton WordNet covers only English, there has been an effort to create WordNets for other languages. It has resulted in the project called EuroWordNet (Vossen and Hirst, 1998), which is a multilingual database based on the same principles as PWN in terms of synsets with basic semantic relations between them. The project was completed in the summer of 1999 and currently is comprised of WordNets for seven other European languages (Czech, Dutch, Estonian, French, German, Italian and Spanish).

Synsets in the WordNets of all listed languages are linked via the interlingual index coming from PWN. The languages are interconnected by means of this index, thus it is possible to traverse from the synsets in one language to similar synsets in any other language. The index also gives us an access to a shared top-ontology of 63 semantic items. This top-ontology provides a common semantic framework for all the languages, while language specific relations are maintained in the individual WordNets. The database can be used in both monolingual and cross-lingual NLP tasks, but in contrast to PWN, some parts of the WordNets are not freely available (for example the Dutch part).

The EuroWordNet project has been followed by similar projects for other languages. In 2001, the BalkaNet project was developed, intended to extend the database of WordNets linked to PWN. It has included a development of Balkan WordNets (Greek, Turkish, Romanian, Bulgarian and Serbian) as well as the second phase of the development of the Czech WordNet (Pala and Smrž, 2004).

1.3.3 *Frames*

Besides representing standalone concepts, linguists must be able to organize lexical units in high-level structures. As for symbolic logic, the basic units are predicates, which are connected by operators to create formulas representing high-level structures. Another possible way of organizing concepts is to use structures such as **frames**. *frame*

Concerning knowledge representation, the frame is a data structure introduced by Marvin Minsky (Minsky, 1975). It is intended to represent complex objects or stereotyped situations. We can think of a frame as a network of nodes and relations, where some nodes are fixed, representing objects that are always true about the frame. The rest of the nodes represent **slots**, which must be filled in with specific instances of data. *slot*

Being inspired by Minsky, many researchers from different branches of science have followed him in the idea of frames. In linguistics, the best known frame-based theory is Charles J. Fillmore's Frame Semantics (Fillmore, 1982), which will be discussed later. It should

2 http://wordnet.princeton.edu/wordnet/download/

be stated that the roots of Fillmore's frames date back to his earlier work The Case for Case (Fillmore, 1968), in which he introduces deep (semantic) cases that are today understood as **semantic roles**.

semantic role

Before we proceed to an in-depth description of frames in later chapters, it is useful to acquire an intuitive notion of semantic frames in linguistics.

Figure 1.1 depicts two semantic frames evoked by lexical units *consume* and *eat*. In this specific interpretation of the concept of

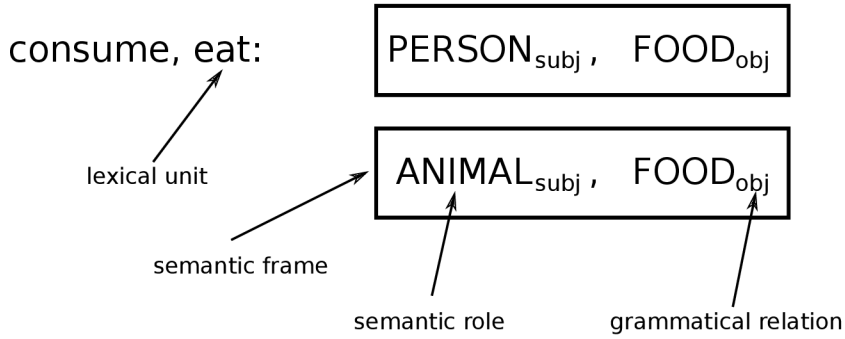


Figure 1.1: Semantic frames evoked by lexical units *consume* and *eat*.

semantic frames, a frame consists of a set of slots, where each slot represents a particular grammatical relation. Here, the slots represent grammatical subject and direct object of lexical units. The slots are filled in with semantic role labels that account for semantic categories of possible grammatical relation realizations.

The frames in the figure distinguish two usage patterns of the words 'consume' and 'eat'. The first frame captures a situation where the subject of the process of eating is a person, the second represents the case where the subject is an animal. Such semantic frames can be used, for example, to describe collocational behavior of lexical units, or to distinguish between different meanings of polysemous words.

1.4 PROBABILISTIC FRAMES

In exact sciences such as mathematics and physics, the most important concepts are described by very simple formulas, e.g. $e = mc^2$. It is not because we are doomed to simplify a very complex world, it is because nature in its roots has simple rules that make up the complex system together. In humanities, by contrast, when describing natural languages, for example, people have not invented any small and comprehensive set of rules yet. An informal and incomplete grammar concept of the English language is described in more than 1700 pages (Quirk et al., 1985). Perhaps it is because natural languages do not have the elegance of physics. Maybe there are some simple regularities that have not been discovered yet. Whatever the true reason is, we should maybe stop searching for comprehensive,

extremely elegant theories, and make use of simple statistics and massive dataset available for free in an electronic form.

It is a well known disagreement between Noam Chomsky and Peter Norvig. "The notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term", wrote Chomsky (1969b). One of his main arguments is that the probability of a novel sentence must be zero, and since novel sentences are generated all the time, there is a contradiction. This argument, however, is overcome by current language models that can assign non-zero probabilities to novel sentences thanks to sophisticated smoothing methods (Manning and Schütze, 1999). Moreover, Norvig argues that in engineering applications of NLP such as search engines, speech recognition or machine translation, the state-of-the-art systems are mostly based on statistical models (Norvig et al., 2009). His complete argumentation can be found on his website[3].

When the Brown Corpus with one million English words was released in 1961, it was the beginning of the statistical natural language processing. A few years later, Kucera et al. (1967) published their classic work providing basic statistics of American English, and the impact of statistics in NLP began to rise. Since then we have seen several notable corpora, and nowadays, we have access to million times larger web corpora, for example at the Linguistic Data Consortium[4]. The world is not black and white, however. The very large corpora are extracted from the Web, thus they contain a lot of spelling and grammatical errors, incomplete sentences and all sorts of other errors. They are not manually annotated with part-of-speech tags nor with any kind of syntactic or semantic information. But their extreme size outweighs all these drawbacks. All we need to know is how to extract useful models from the data.

1.4.1 *Practical Semantic Frames*

Databases of semantic frames can basically serve two different purposes – linguistic research and natural language processing. Most of the current semantic frame databases are based on formalisms proposed by linguists with the primary goal to describe natural languages. When we are focusing on practical application in NLP, however, we require quite different qualities from the semantic frames.

Let us start by listing properties that are typical for current semantic frame databases and are not desirable in NLP. Afterwards, we will proceed to the proposal of the ideal form of semantic frames for natural language processing.

3 http://norvig.com/chomsky.html
4 http://www.ldc.upenn.edu/

The most limiting property of current lexicons of semantic frames is that they are created manually by highly trained linguists. The complete manual work is recently supported by some sort of semi-automatic preprocessing but it still requires a non-trivial human effort. This results in several negative consequences:

- **Creation of them is very time-consuming and expensive.** For example the development of FrameNet started in the end of 1990s and the work is still in progress. In fact, it is a never-ending story because new words that need to be assigned to proper semantic frames steadily arise. As far as I know, the expenses covering annotators' salary have not been published, but it must be a huge amount of money.

- **Coverage of the resources is usually small or limited to some specific domain**. According to the information provided on its website[5], the largest database of semantic frames, FrameNet, currently covers 12,711 lexical units. In comparison, the Second Edition of the Oxford English Dictionary (Simpson and Weiner, 1989) contains approximately 153,000 entries for nouns, adjectives and verbs in current use, and more than 47,000 obsolete words. It means that FrameNet contains less than one twelfth of the currently used vocabulary of the English language.

- **Most of the resources do not provide any information about relative frequency of usage in corpora.** When dealing with different meanings or frames of a lexical unit, it is very useful to know the importance of the meanings. For instance, both patterns `[Person]` *acquire* `[Physical_object]` and `[Person]` *acquire* `[Disease]` reflect a correct usage of the verb 'acquire', but the former is much more frequent in English. Similarly, it is worth knowing, for instance, that the semantic role `Person` is more frequently used than the semantic role `Teacher` in English semantic frames in general. This information can be used, for example, as a prior probability of semantic frames and roles in the task of semantic roles labeling.

- **Notion of semantic classes and frames is subjectively biased when the frames are created manually without corpus evidence.** If two people are asked, for example, to create a frame for arriving, it is very likely that they define different set of semantic roles. Let us have a look at the frame `Arrive` in FrameNet. There are two core frame elements and seventeen non-core frame elements, including `Frequency`, defined as "How often the Theme arrives at the Goal". It is meaningful because we can count times of arriving. However, why is `Frequency` missing in other frames describing countable events,

5 https://framenet.icsi.berkeley.edu/fndrupal/current_status

such as `Arrest`, `Control`, `Hiring`, `Travel`, etc.? Human annotators make mistakes and have different opinions. It is impossible to avoid inconsistencies and errors even if a very accurate and well-written annotation manual is available.

All the drawbacks we listed so far stem from the manual creation of semantic frames. If we want to avoid them and develop semantic frames that would be easily applicable in practical NLP, we need to propose a method for creating semantic frames **automatically** by using machine learning techniques.

The biggest successes in natural language related machine learning applications in recent years have been achieved in statistical machine translation and statistical speech recognition (Norvig et al., 2009).The reason, however, is not that these tasks are easier than other tasks in NLP. In fact, they are much harder. The true reason is that we have big training data available. Every day the European Union clerks generate a huge amount of texts that are required to be translated to all EU languages. This data is collected in order to compile big parallel corpora. Similarly, speech transcriptions are routinely done for real human needs.

On the other hand, we do not have large enough training data for traditional NLP tasks such as parsing or named-entity recognition. This data needs to be created on purpose by hired annotators.

An important lesson we should learn from history is that the usage of large and naturally created data is better than hoping for better-fitting but manually annotated data sets which are usually much smaller. Since no examples of semantic frames are generated naturally in large amounts, we are doomed to exploit **unsupervised** techniques on large text corpora that are not semantically annotated.

When creating a database of semantic frames manually, the author has to decide how many semantic frames are required to define. In other words, he or she needs to know what the intended granularity of senses is. The same problem also arises with the choice of the proper number of semantic roles for each semantic frame. It is obvious that the number of frames and roles increase with the growing size of data which the frames are required to describe. Since we want to learn semantic frames automatically from corpora, we need to use a **non-parametric** method for discovering the best number of frames and roles based on the training corpus.

Current lexicons of semantic frames usually do not have any weights of importance for the valid frames of a lexical unit. The weights are important for integrating the semantic frames into a statistical NLP system. That is why it is reasonable to develop **probabilistic** semantic frames so that there is a probability distribution over the set of semantic frames for every lexical unit and a probability distribution over valid realizations for every semantic role. The probabilistic

semantic frames enable us to make it easier to annotate a corpus with the frames automatically and integrate them into a statistical system.

The largest database of semantic frames, FrameNet, has two important properties that are not very useful for practical NLP. First, the set of semantic roles is frame-specific, and secondly, the roles are not explicitly connected with grammatical relations.

The former property implies that we are unable to investigate relationships between semantic roles of different frames. For example, we do not know from FrameNet that the semantic role Self_mover in frame `Self_motion`, defined as "Self_mover is the living being which moves under its own power", is of a related type to Traveler in frame `Travel`, defined as "This is the living being which travels". Specifically, we do not have any explicit information about semantic similarity of these two semantic roles other than the textual description. In order to be able to compare semantic roles from different semantic frames, it is convenient to **share the set of semantic roles** across all semantic frames. The probability distribution over the semantic role realizations connected with every semantic role then enables us to compute similarity between two different semantic roles.

The latter mentioned property of FrameNet – that the semantic roles are not explicitly connected with grammatical relations – is double-edged. On one hand, as we will see in section 2.2, some semantic roles can be realized in different grammatical relations and the assignment would be ambiguous. It is most likely the reason why grammatical relations are not included in FrameNet frames. This problem, however, can be solved by introducing more frames for a lexical unit. On the other hand, if they were a component of semantic frames, the automatic annotation of texts with semantic frames would be much easier, which is crucial for practical NLP. That is why it is desirable to **incorporate the information about grammatical relations** into the proposed model of semantic frames.

To conclude, let us list the most important properties of a newly proposed model of semantic frames:

- semantic frames must be automatically generated from text corpora;
- algorithm must be unsupervised;
- model should be probabilistic;
- the model should provide a possibility to estimate its parameters automatically;
- set of semantic roles should be shared between different semantic frames;
- information about valid grammatical relations should be included.

In order to create semantic frames that fulfill our requirements, we need to solve three orthogonal problems – choosing a representational language, encoding a model in that language, and performing inference on the model. These challenges will be discussed in the second part of the thesis.

1.4.2 *Related Work*

Statistical unsupervised approaches to semantic parsing have recently acquired the attention of many researchers. Most of the approaches related to the semantic frame induction focus on modeling selectional preferences and semantic role labeling.

A pioneer approach in this area was a model proposed by Rooth et al. (1999), who casted the problem of selectional preference induction into a probabilistic latent variable framework. Each observed predicate argument is generated from a latent variable, and this latent variable is generated from the underlying distribution over variables. The discovery of the latent variables, representing semantic clusters of the predicate argument realizations, is carried out using the expectation-maximization procedure.

Following Rooth's work, Durme and Gildea (2009) and Séaghdha (2010) proposed several models for selectional preference induction based on Latent Dirichlet Allocation. Their models separately learned selectional preferences for verb-object, noun-noun, and adjective-noun relations. The generative models sample both predicate and argument realizations together.

In order to model multiple selectional preferences, Ritter et al. (2010) developed LDA-SP. It models the subject and object grammatical relations together. The model is based on an extension of Latent Dirichlet Allocation known as LinkLDA (Erosheva et al., 2004), which simultaneously models two sets of distributions over topics. These two sets represent the two arguments for the relations.

A non-trivial amount of work have been done in the area of unsupervised semantic role annotation for PropBank (Swier and Stevenson, 2004; Grenager and Manning, 2006; Lang and Lapata, 2010, 2011a,b; Titov and Klementiev, 2012). These models are, however, PropBank-specific, and thus far away from the goal of designing probabilistic semantic frames without relying on manually-created lexical resources.

Recently, Modi et al. (2012) published the description of a framework modeling both semantic frame and role representations together, which is the closest approach to the goals stated in this thesis. They proposed a Bayesian non-parametric method, which uses the hierarchical Pitman-Yor process (Pitman, 2002) to model the statistical dependencies between predicate and argument clusters, as well as distributions over lexical realizations of each cluster. The primary

goal of Modi's work is to generate FrameNet-like semantic frames and roles. Similarly to FrameNet, however, the semantic roles are treated as frame-specific, and thus the model does not try to discover any correspondence between roles in different frames.

To the best of my knowledge, the first unsupervised model that generates semantic frames with a shared set of semantic roles and fulfills all requirements stated in this thesis is called LDA-Frames. It has been published in Materna (2012a, 2013) by myself and will be described in details in this thesis.

1.5 STRUCTURE OF THE THESIS

This Ph.D. thesis is divided into two logical parts. The first part, which started with this opening chapter, is dedicated to introducing the reader to basic concepts and ideas behind semantic frames and related linguistic constructions. It also provided a motivation for constructing lexicons of probabilistic semantic frames. The following chapter goes deeper into the issues of semantic frames and verb valencies. It introduces semantic roles, elementary constituents of the frames, and their role in lexical semantics. Within this section, the reader also explores how the semantic frames are connected with syntax and what motivation leads us to build lexicons of semantic frames. The first part is enclosed by a list of several existing lexicons of semantic frames and verb valencies, along with their brief description.

The core of this work is described in the second part. Since the suggested model for semantic frames is of a stochastic matter, the thesis provides the reader with a description of probabilistic graphical models and probability distributions required for the models. An important component of the proposed probabilistic semantic frames are topics models. This is the reason why Latent Semantic Analysis, Probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation are presented afterwards. The probabilistic semantic frame model itself is called LDA-Frames and is described in detail along with its non-parametric extension called Non-Parametric LDA-Frames, a hyperparameter estimation method, and a parallelized algorithm which enables us to generate the probabilistic semantic frames from huge data sets in a distributed manner. The model is subsequently evaluated using several standard methods. At the end of the thesis, the model is tested on two applications. The first is the construction of a lexicon of semantic frames based on the learned model, which is compared with a manually created lexicon from the CPA project. The second is the usage of LDA-Frames for building a thesaurus.

2 SEMANTIC FRAMES FOR NATURAL LANGUAGE PROCESSING

Robust morphological analyzers and syntactic parsers are widely used in many natural language processing applications, and their quality has been steadily increasing in recent years. The analysis of morphology and syntax, however, covers only a small portion of language information needed for complete understanding of the sentences that are parsed. Therefore, a semantic parsing must be employed. It is apparent that some sort of semantic information is even required for both morphological and syntactic disambiguation. Take for instance the example below

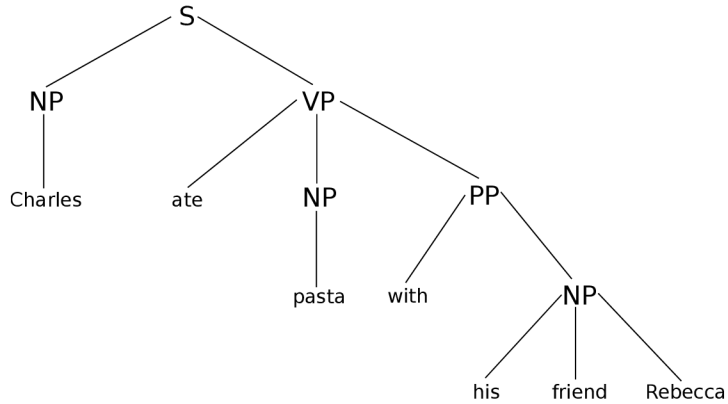
(2.1) *Charles ate pasta with his friend Rebecca.*

This sentence can be interpreted in two different ways. At first reading (figure 2.1a), the prepositional phrase 'with his friend Rebecca' is attached directly to the verb phrase. It means that Charles is likely sitting at the table with Rebecca and eating his pasta. In the second reading (figure 2.1b), however, the prepositional phrase is attached to the noun phrase representing the direct object of the verb 'ate'. This parsing may be interpreted as Charles is eating his friend Rebecca with pasta. Both interpretations are grammatically correct, but the second is semantically highly implausible. This problem is known as **prepositional phrase attachment** (Merlo and Ferrer, 2006).

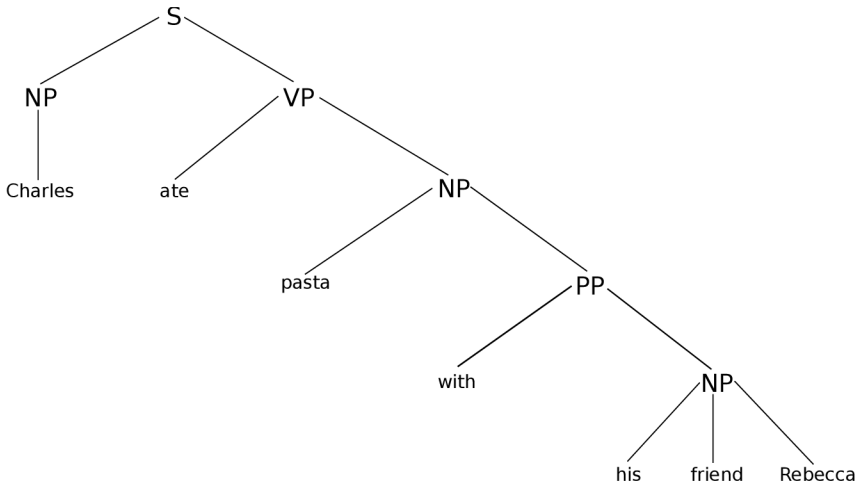
prepositional phrase attachment

In order for us to be able to analyze such sentences correctly, we need to identify participating entities, their type, their relation to each other, and especially their relation to the verb predicate. In this particular example, there is one verb predicate 'ate' and three related entities – 'Charles', 'pasta' and 'friend Rebecca'. The construction of the sentence and the entity's relation to the predicate assigns it a specific role. 'Charles' is the entity responsible for carrying out the action, 'pasta' is the entity affected by the action of 'Charles' and 'friend Rebecca' is accompanying 'Charles' in the process. These roles have various names in the literature, e.g. semantic roles (Givón, 1990), deep semantic cases (Fillmore, 1968) and thematic roles (Dowty, 1986), which might be confusing. We will use the term **semantic roles** for the rest of this thesis.

When analyzing the semantic roles and their connection to predicates, we are able to link them to an ontology. It can help us recognize that, given the verb 'eat', the affected entity should be food and that 'Rebecca' is not a typical instance of food. Using this kind



(a) A prepositional phrase is attached to the verb phrase.



(b) A prepositional phrase is attached to the noun phrase.

Figure 2.1: Example of the Prepositional Phrase Attachment problem.

of information, we can solve the problem of prepositional phrase attachment from example (2.1).

2.1 SEMANTIC ROLES

The roots of the concepts of a Semantic Role (SR) go back to Fillmore's book on deep semantic cases The Case for Case (Fillmore, 1968). In this book, Fillmore was the first to ask the question about evidence for the existence of semantic roles as semantically typed arguments of verbs. In his conception, every noun phrase associated with a verb is assigned a deep semantic case. The originally published list of deep cases included Agentive, Dative, Instrumental, Factitive, Locative and Objective, where the number and type of the deep cases are determined by the verb itself. In the English language, verbs are typically intransitive (require one argument), transitive (two arguments) or ditransitive (three arguments). For example, the verb 'love' requires Agentive and Objective, so it is transitive.

During many years of research, the deep semantic cases evolved to a new list of semantic roles, which is generally respected by linguists (Saeed, 2009). The most important semantic roles with their description and examples are listed below:

AGENT is the initiator of an action, capable of acting with volition. It is usually the grammatical subject of the verb in an active clause. For example

(2.2) ***David*** *eats bananas and porridge for breakfast.*

(2.3) ***The government*** *locked the unused reactors for 35 years.*

PATIENT is the entity affected by some action and often undergoing some change of state. It is usually the grammatical object of the verb in an active clause. For example

(2.4) *That child broke* ***the window****.*

(2.5) *The barber cut* ***his hair****.*

THEME is the entity affected by some action, which is moved or whose location is described. Its state, however, is not changed. It is usually the grammatical object of the verb in an active clause. For example

(2.6) *I put the book on* ***the shelf****.*

(2.7) *John has* ***two children****.*

EXPERIENCER is the entity which perceives an action but which is not in control of the action or state. For example

(2.8) ***He*** *listened to the radio.*

(2.9) ***The pupils*** *saw their new teacher.*

BENEFICIARY is the entity for whom a benefit action is performed. For example

(2.10) *They helped* ***Japan people*** *after the earthquake.*

(2.11) *The parents rented an apartment* ***for their son.***

INSTRUMENT is the means by which the action of an agent is performed. Grammatically, it is usually a prepositional phrase. For example

(2.12) *Martin smashed the car with* ***a big hammer.***

(2.13) *Two copies of the agreement have been signed with* ***the gold pen.***

LOCATION is the place where something is situated or takes place. For example

(2.14) *The children played soccer* ***on their playground.***

(2.15) *He found the cell phone* ***under the couch.***

GOAL is the entity where the action is directed towards, in both literal and metaphoric meaning. For example

(2.16) *Allan gave all his money* ***to the croupier.***

(2.17) *He told the joke* ***to his friends.***

SOURCE is the entity where the action is directed from, in both literal and metaphoric meaning. For example

(2.18) *He came to Prague* ***from the United States.***

(2.19) *We heard the story* ***from his friend.***

STIMULUS is the entity causing an effect in the experiencer. For example

(2.20) *She did not smell* ***the smoke.***

(2.21) ***The noise*** *frightened the people.*

The difference between the semantic roles may not always be apparent. Thanks to Jackendoff (1974), distinguishing between some semantic roles is more straightforward. For example, to differentiate Agent from Patient, we can ask the question *What did X do?*, where 'X' is replaced with a potential Agent. If the question can be applied, the assumption is correct. Similarly, Patient can be tested by using questions as *What happened to Y?* or *What did X do to Y?*.

The situation is becoming more complicated when we try to distinguish between Patient and Theme. The general definition says that Patient undergoes a change, whereas Theme simply changes the location or is located. This seems straightforward, but some examples might be quite complicated. Let us have a look at the following sentence:

(2.22) *The blizzard covered the road with snow.*

Has the road undergone some change of state (and so it is classified as Patient) or not? Moreover, recognizing Agent in the sentence (2.22) is unclear as well. The blizzard is hardly acting intentionally, so it does not fulfill the definition of Agent. That is why Cruse (1986) divided Agents into several subtypes based on intentionality and volition of the roles. There are many other examples of sentences with role assignments being unclear or ambiguous, and the problem of identifying semantic roles is still not settled in a satisfactory way.

Besides the unclear role assignment, there is another difficult question of whether a single entity can fulfill two or more semantic roles and whether the situation where some noun phrases have no role assignment is correct. This issue has been studied by Chomsky (1993) who stated that there must be a one-to-one correspondence between noun phrases and semantic roles in a sentence. This is known as Chomsky's Theta-Criterion. By contrast, Jackendoff (1985) discovered some counter-examples that deny the theory. These issues are still under investigation and are not settled.

2.2 CONNECTION WITH SYNTAX

As one can see from the previous examples, in English there is a tendency of having some semantic roles to appear as realizations of specific grammatical relations. For example, Agents tend to be subjects, Patients and Themes tend to be direct objects, and Instruments or Locations tend to be prepositional phrases. Unfortunately, this is not always the case. There are generally two reasons for this (Saeed, 2009). The first comes when the roles are simply omitted and the grammatical relations shift to react to this. The second case happens when the speaker chooses to alter the usual matching between roles and grammatical relations. This is usually marked by an accompanying change of verbal voice. It is illustrated in following sentences:

(2.23) ***John*** *broke the window.*

(2.24) ***I*** *felt tired after running ten kilometers.*

(2.25) ***The car*** *crashed.*

(2.26) ***The key*** *opened the lock.*

In example (2.23), the bolded subject is Agent, in example (2.24) the subject is Experiencer, in example (2.25) the subject is Patient, and the last example (2.26) is the case where the subject is Instrument.

Even though the mapping between grammatical relations and semantic roles is not straightforward, the syntactic parsing is an important part of the algorithms for semantic role labeling. **Semantic role labeling** is a task where semantic roles are assigned to words

semantic role labeling

or phrases in natural languages. The parse tree as an input for machine learning is required, for example, in approaches to semantic role labeling by Gildea and Jurafsky (2002) or Pradhan et al. (2005). Moreover, there are some lexical resources where the grammatical relations play a key role, and are stored together with semantic roles in the valency frame, as in the lexicon by Hlaváčková and Horák (2006). Grammatical relation identification is also required for generating probabilistic semantic frames called LDA-frames, which are the main result of this thesis.

2.3 VERB VALENCIES AND SEMANTIC FRAMES

As we saw earlier, verbs usually have particular requirements for their semantic roles. Since the information about prototypical behavior of verbs in sentences is a part of people's knowledge about a verb, it is reasonable to store such information in lexicons of **verb valencies**.

verb valency

The linguistic interpretation of valences is derived from the definition of valency in chemistry and was used for the first time by Lucien Tesniére in his work about syntactic analysis of sentences (Tesnière, 1959). In the generative grammar literature, e.g. Gruber (1965), the frames of semantic roles are called **thematic role grids** or **theta grids** for short. Nowadays, there are plenty of various models of verb valencies, each having its own uniquenesses, but we can roughly say that the valencies are verb arguments required by the verb in a particular language. For the rest of this thesis, we will call them **valency frames** or **semantic frames**.

thematic role grid

valency frame

semantic frame

A simple example of valency frame in English can be

(2.27) **feed**: <Agent, Patient>

This frame tells us that the verb 'feed' has two obligatory semantic role arguments (the verb is transitive), where one of them is Agent and the second is Patient.

2.3.1 *Obligatory vs. Facultative Valencies*

Some frames may capture both obligatory valencies that are required in all circumstances, and facultative valencies, which are optional. The information about the type of valency, however, should be presented in the frame. For instance let us have a look at the following sentences:

(2.28) [Michel]$_{\text{Agent}}$ **put** [the book]$_{\text{Theme}}$ [in the library]$_{\text{Location}}$.

(2.29) [Michel]$_{\text{Agent}}$ **read** [the book]$_{\text{Theme}}$ [in the library]$_{\text{Location}}$.

The bolded predicates in sentences (2.28) and (2.29) capture the same valency structure

(2.30) **put, read**: <Agent, Theme, Location>.

However, Location in the former example is obligatory and in the latter facultative. That is why it would be better to distinguish the type of valency explicitly

(2.31) **put**: <Agent^{obl}, Theme^{obl}, Location^{obl}>

(2.32) **read**: <Agent^{obl}, Theme^{obl}, Location^{fac}>.

A deeper insight into the issues of obligatoriness and facultativeness can be found in Radford (1988) or Haegeman (1994). In general, it is assumed that all verbs may co-occur with adverbials of time, manner and place, therefore, such arguments are usually not listed in lexicons of verb valencies.

2.3.2 *Verb Classes*

Identifying valency frames soon led to grouping verbs with the same frames into classes. For example, the English verbs 'give', 'pay', 'supply', 'transfer', etc. may form the class `Transfer`. All those verbs capture the situation where Agent transfers Theme to Recipient, which is encoded using the following valency frame:

(2.33) **give, pay, supply, transfer**: <Agent, Theme, Recipient>.

An example of more elaborated verb classes are Beth Levin's verb classes and alternations (Levin, 1993). In her work, Levin recognizes approximately 80 types of alternations. She argues that syntactic variations are a direct reflection of the underlying semantics. An example of such an alternation is the causative/inchoative alternation. The causative (transitive) alternation is a change of state where Agent is explicitly mentioned, and the inchoative (intransitive) describes the change without a reference to the causer. This alternation is illustrated in the following example, where the sentence (2.34) represents the causative usage of verb 'break' and sentence (2.35) the inchoative usage:

(2.34) *John crashed the car.*

(2.35) *The car crashed.*

The alternations which are (or are not) applicable to a given verb then determine its semantic class.

2.3.3 *Verb-specific vs. General Semantic Roles*

One of the most problematic parts of creating verb valency lexicons is the identification of the set of semantic roles. We introduced a list of the most important semantic roles in section 2.1, but there is still a continuous discussion between linguists about how the ideal set of semantic roles should look. On one hand, a small set of general semantic roles has an advantage of simplicity and universality. On the other hand, a larger set of semantic roles allows distinguishing small distinctions between verb argument meanings.

In the previous section we revealed that Patient is defined as the entity affected by an action of Agent. However, for example Dixon (1991) remarked that there are many types of affectedness. The intensity of affectedness is illustrated in following examples:

(2.36) *Mary **touched** the window with her finger.*

(2.37) *John **pushed** the door.*

(2.38) *Charles **squeezed** the ball in his hands.*

(2.39) *Peter **smashed** the wall with a hammer.*

The intensity of affectedness varies between touching and smashing the Patient role instantiation.

Regardless of problems arising with an extremely small number of shared semantic roles on one hand, or with verb-specific sets of semantic roles on the other hand, both approaches are used in practice as is shown in the frames below:

(2.40) **attack**$_{\text{VerbNet}}$: `<Agent, Theme>`

(2.41) **attack**$_{\text{FrameNet}}$: `<Assailant, Victim>`

Example (2.40) illustrates valency frame of verb 'attack' in VerbNet, which uses a general set of semantic roles, example (2.41) shows the frame taken from FrameNet, which is based on lexical-unit-specific semantic roles.

2.3.4 *Relations Between Verb Valencies and Semantic Frames*

So far, we have discussed only valencies of verbs. A similar idea, however, can be applied to other parts of speech, specifically nouns, adjectives and adverbs. While verbs usually require some arguments obligatorily, nouns, adjectives and adverbs evoke prototypical situations, where the semantic arguments are predominantly facultative. The prototypical situations can also be described by similar frame structures as the verb valency frames. According to Fillmore (1982),

we generalize the idea of verb valencies and call them semantic frames.

To illustrate how semantic frames may look, let us list some examples of semantic frames for lexical units of various parts of speech taken from FrameNet together with their semantic role description:

(2.42) **reason**: <Action, Agent, State_of_affairs>

Action []	The action that the Agent performs in response to a State_of_Affairs. I find that all my REASONS for postponing giving birth still apply
Agent []	The person who responds to a State_of_Affairs by performing some Action. Those are my REASONS , " said Toby .
State_of_Affairs []	The eventuality that motivates the Agent"s performing a particular Action in response to it. My REASON *is* that our cities are now designed round the motor car

(2.43) **lucky**: <Protagonist, Role, State_of_affairs>

Protagonist [pro]	The Protagonist is the one for whom the destiny is evaluated. He is LUCKY.
Role [rol]	The Role filled by the Protagonist. He is a LUCKY gambler.
State_of_affairs [eve]	The State of affairs that is evaluated. That was a LUCKY catch.

(2.44) **first**: <Experience>

Experience [exp]	An event happens for the first time within the Context. Made from flax, linen was FIRST used by the ancient Egyptians. The Cobra was FIRST deployed to Vietnam in September 1967.

The basic idea proposed by Fillmore (1982) is that a semantic frame captures typical, not necessarily obligatory, semantic participants in natural language sentences. The point is that a single word does not provide full semantic information about its meaning. When a

word has multiple meanings, every meaning should be associated with a different semantic frame. Thus, we can see semantic frames as a generalization of valency frames, which connects semantics with syntax and can be applied to lexical units of various parts of speech. More details about Fillmore's semantic frames will be described in the subsequent chapter.

2.4 MOTIVATION FOR IDENTIFYING SEMANTIC FRAMES

Semantic frames have primarily been proposed to serve linguists to understand the language (Fillmore, 1982), however, we can list several practical applications from the natural language processing field. The main purpose of identifying semantic frames and semantic roles is the linkage of the lexical units' syntactic behavior with their semantics. Their usage for the task known as word sense disambiguation is very straightforward.

Word Sense Disambiguation (WSD) is a natural language processing task whose goal is to identify the correct word sense for a given word if the word has multiple meanings. In order to be able to find the correct word sense, the context (such as a sentence or paragraph) in which the word is used has to be provided. The set of possible word senses is usually predefined. If we do not know the list of senses, the task changes to Word Sense Induction (WSI).

There are basically two approaches to WSD – a shallow approach and a deep approach. Shallow approaches just consider the surrounding words and do not try to understand the text. For example, have a look at following sentences:

(2.45) *A lot of fish were hidden near the **bank** of the river.*

(2.46) *A **bank** account is a financial account between a bank customer and a financial institution.*

There are two occurrences of the word 'bank' with different meanings, which can easily be distinguished only based on the surrounding words. In example (2.45), words like 'fish' and 'river' indicate that the correct meaning should be a *border of a river*. In sentence (2.46), the correct meaning of 'bank', which is a *bank account*, can be detected by the occurrence of words 'account', 'financial', 'customer' and 'institution'. This approach, while theoretically not much powerful, works quite well in practice.

Deep approaches are focused on the semantics of languages and on the knowledge of the world. There are many sentences where the shallow approach fails and we need to incorporate a deeper knowledge. For example in sentence

(2.47) *The company constructed the **bank** near the river.*

The presence of word 'river' in the sentence might indicate that the correct word sense is *border of a river*, but the world knowledge tells us that *building* is more likely being constructed near the river than *border of a river*.

For the deep understanding of the meaning, we usually require some kind of lexical resource in a computer-readable format describing semantics of ambiguous words. Semantic frames are examples of such lexical resources and are indeed widely used for WSD (Ide and Véronis, 1998; Dang, 2004; Navigli, 2009). They can be used as selectional restrictions disambiguating the correct sense based on the semantics of predicate's arguments. It means that one semantic frame should not represent multiple meanings of one word. For instance, let us return back to sentence (2.45). One of the possible semantic frames of predicate 'to construct' is

(2.48) **construct**: <Institution$_{subj}$, Building$_{obj}$>.

If we are able to map semantic roles to their grammatical realizations in the sentence (Institution to 'The company' and Building to 'bank'), we can ask whether *building* is more likely an instance of semantic role Building than *border of a river*, and select the best-fitting meaning of 'bank'.

Word sense disambiguation is an important part of many complex tasks such as machine translation, information retrieval, information extraction, etc. That is why the semantic frames may be useful in a lot of fields of natural language processing.

Another NLP application of the semantic frames, which is quite different from WSD, is identifying semantic classes of words or building thesauri. For example, both verbs 'learn' and 'teach' represent an action of transferring knowledge, however, in case of 'learn', the agent is receiving knowledge, in case of 'teach', the agent is giving knowledge. These two directions of transferring knowledge may be considered to represent different semantic classes. It can simply be identified by looking at semantic frames for 'learn' and 'teach'

(2.49) **learn**: <Student, Subject>

(2.50) **teach**: <Teacher, Subject>

Apart from identifying semantic classes, the comparison of semantic frames can be used for building a thesaurus. A thesaurus is generally a lexical resource grouping together words or phrases according to the similarity of their meanings. This similarity need not only be defined in terms of synonymy. Antonymy, hyponymy, hypernymy and week synonymy are also valid semantic relationships for grouping words together in some types of thesauri.

The basic idea behind building thesauri using semantic frames is that words with similar meanings should share most of their valid semantic frames. An example of building thesaurus based on semantic frames can be found in Materna (2012b).

3 EXISTING FRAME-BASED LEXICAL RESOURCES

Nowadays, several important lexical resources providing extensive databases of semantic frames or verb valencies for English exist. The set of the three largest and best-known databases is comprised of PropBank, VerbNet and FrameNet. Even though the mentioned lexical databases capture semantic frames or verb valencies that have been described in the previous chapter, they differ in many aspects and details. All these lexical databases together with three representants of the Czech language – BRIEF, Vallex and VerbaLex – and several examples of semantic frame databases for other languages are described in this chapter.

3.1 PROPBANK

The primary goal of developing the Propositional Bank[1] (or ProbBank for short) is to annotate the Penn Treebank[2] phrase structures with predicate-argument relations (Kingsbury and Palmer, 2002; Kingsbury et al., 2002).

For each verb, correct word senses are identified. Also, for each sense, the argument structures are stored. The semantic role labels are chosen to be generic and neutral, so they are distinguished only by numbers (for example Arg0, Arg1, Arg3, etc). It is important that the numbers may have different meanings with different verbs. One of the goals, however, is to provide consistent argument labels across different syntactic realizations with the same verb. For example

(3.1) [John]$_{Arg0}$ **broke** [the window]$_{Arg1}$.

(3.2) [The window]$_{Arg1}$ **broke**.

The second task for the PropBank annotation involves assigning functional tags to all modifiers of the verb. The modifiers are denoted as ArgM and are further subclassified based on the type, such as manner (MNR), time (TMP), location (LOC), purpose (PRP), etc. Finally, the PropBank annotation involves finding antecedents for empty arguments of the verbs. A simplified example of annotations is shown in figure 3.1.

1 http://verbs.colorado.edu/~mpalmer/projects/ace.html
2 http://www.cis.upenn.edu/~treebank/

Texas Instruments Inc. opened a plant in South Korea to manufacture control devices.

```
Arg0:      Texas Instruments Inc.
REL:       opened
Arg1:      a plant
ArgM-LOC:  in South Korea
ArgM-PNC:  to manufacture control devices
```

Figure 3.1: PropBank example entry for lexical unit `'open'`.

3.2 VERBNET

VerbNet[3] is a lexicon of English verbs developed at the Department of Linguistics, University of Colorado. It is partially based on the PropBank project – it uses its syntactic frames; see for example Dang et al. (1998); Kipper et al. (2000); Schuler (2005); Kipper et al. (2008). The main difference in comparison to PropBank is that VerbNet enhances the frames with additional semantic information. The fundamental assumption is that the syntactic frames of a verb directly reflect the underlying semantics. VerbNet associates the semantics of a verb with its syntactic frames in terms of combining them with traditional lexical semantic information such as thematic roles and semantic predicates.

Each syntactic frame in VerbNet is assigned to a semantic class based on Beth Levin's verb classification (Levin, 1993). The alternations which are (or are not) applicable to a given verb then determine its semantic class. Each VerbNet class contains a set of syntactic descriptions, or syntactic frames, depicting the possible surface realizations of the argument structure. Unfortunately, the classes are problematic because they do not take into consideration corpus data. It can be observed that the individual classes contain verbs that considerably differ in their meanings, thus the classes are sometimes semantically inconsistent.

The lexicon has recently been extended with an additional set of new classes (approximately 200 additional classes), and currently is the most comprehensive, versatile and freely available resource of Levin-style verb classification for English.

3.3 THE BERKELEY FRAMENET

The linguistic basis of the frame semantics and the FrameNet project can be found in the theory of case grammar, beginning with the work by Charles J. Fillmore (Fillmore, 1968). It was offered as a contribution to the transformational-generative grammar promoted

3 http://verbs.colorado.edu/~mpalmer/projects/verbnet.html

by Noam Chomsky and his followers. The main idea of the case grammar consists of the proposal that deep syntactic structures are best described as configurations of **deep cases**. These deep cases are expressed as a fixed set of general semantic role names such as Agent, Patient, Time, etc. Since the beginning, however, there have been questions about the correct set of semantic roles, and whether a small and fixed set of semantic roles can characterize all predicates of natural languages.

deep case

Indeed, in his later work, Fillmore (1976, 1977b) showed that such a fixed set of semantic roles is not sufficient to characterize all language predicates, and proposed the theory of **frame semantics** (Fillmore, 1982). The central idea of the frame semantics is that the word meaning is described in a relation to a **semantic frame**, which consists of a target **lexical unit** (pairing of the word with a sense), **frame elements** (its semantic arguments) and relations between them.

frame semantics
semantic frame
lexical unit
frame element

Since the end of 1990s, at the International Computer Science Institute in Berkeley, computational lexicography project FrameNet (Ruppenhofer et al., 2006) has been going on. The main goal of the project is to extract information about the linked semantic and syntactic properties of English words from large electronic text corpora using both manual and automatic procedures. The name "FrameNet" is inspired by WordNet, reflecting the fact that the project is based on the theory of frame semantics and that the semantic frames form a network. The information about words and their properties is stored in an electronic lexical database. Possible syntactic realizations of the semantic roles associated with frames are exemplified in annotated FrameNet corpus.

Currently, the Berkeley FrameNet consists of more than 10,000 lexical units across various parts-of-speech, associated with more than 800 frames exemplified in more than 135,000 annotated sentences.

3.3.1 *Semantic Frames*

The semantic frame is defined by the author as a coherent structure of related concepts such that, without knowledge of all of them, one does not have complete knowledge of one of the others, and are in that sense types of gestalt (Fillmore, 1977a). In other words, it is a script-like conceptual structure that describes a particular type of situation, object or event along with its participants and properties (Ruppenhofer et al., 2006).

A lexical unit in the frame semantics is defined as the pairing of a word with its meaning. Typically, each sense of a polysemous word belongs to a different semantic frame. For example, the `Commerce_sell` frame describes a situation in which a `seller` sells some `goods` to a `buyer`, and is evoked by lexical units such as 'auction', 'retail', 'retailer', 'sale', 'sell', etc.

3.3.2 *Frame Elements*

frame element

The semantic valencies of a lexical unit are expressed in terms of the kinds of entities that can participate in frames of the type evoked by the lexical unit. The valencies are called **frame elements**. The Frame Element (FE) bears some resemblance to the argument variables used in first-order predicate logic, but has important differences derived from the fact that frames are much more complex than logical predicates (Fillmore et al., 2003). In the example above, the frame elements include Seller, Goods, Buyer, etc.

core frame element

peripheral frame element

extra-thematic frame element

FrameNet distinguishes three types of frame elements – **core** FE (the presence of such FE is necessary to satisfy a semantic valence of a given frame), **peripheral** FE (they are not unique for a given frame and can usually occur in any frame, typically expressions of time, place, manner, purpose, attitude, etc.), and **extra-thematic** FE (these FEs have no direct relation to the situation identified with the frame, but add new information, often showing how the event represented by one frame is a part of an event involving another frame). An example of a FrameNet frame is shown in Figure 3.2.

Definition:

An Ingestor consumes food or drink (Ingestibles), which entails putting the Ingestibles in the mouth for delivery to the digestive system. This may include the use of an Instrument. Sentences that describe the provision of food to others are NOT included in this frame.

The wolves DEVOURED the carcass completely.

FEs:

Core:

Ingestibles [Ingible]	The Ingestibles are the entities that are being consumed by the Ingestor.
Ingestor [Ing] **Semantic Type:** Sentient	The Ingestor is the person eating or drinking.

Non-Core:

Degree [Degr] **Semantic Type:** Degree	The extent to which the Ingestibles are consumed by the Ingestor. The wolves DEVOURED the carcass completely
Duration [Dur]	The length of time spent on the ingestion activity. They 've been EATING for hours !
Instrument [Ins] **Semantic Type:** Physical_entity	The Instrument with which an intentional act is performed.
Manner [Manr] **Semantic Type:** Manner	Manner of performing an action.
Means [Mns] **Semantic Type:** State_of_affairs	An act performed by the Ingestor that enables them to accomplishes the whole act of ingestion. The thing ATE by snaking its long tongue out and grabbing with it.
Place [Place] **Semantic Type:** Locative_relation	Where the event takes place.
Purpose [pur] **Semantic Type:** State_of_affairs	The action that the Ingestor hopes to bring about by ingesting.
Source [Src] **Semantic Type:** Source	Place from which the Ingestor takes the Ingestibles
Time [Time] **Semantic Type:** Time	When the event occurs.

Lexical Units:

breakfast.v, consume.v, devour.v, dine.v, down.v, drink.v, eat.v, feast.v, feed.v, gobble.v, gulp.n, gulp.v, guzzle.v, have.v, imbibe.v, ingest.v, lap.v, lunch.v, munch.v, nibble.v, nosh.v, nurse.v, put away.v, put back.v, quaff.v, sip.n, sip.v, slurp.n, slurp.v, snack.v, sup.v, swig.n, swig.v, swill.v, tuck.v

Figure 3.2: FrameNet frame example: `Ingestion`.

3.3.3 *FrameNet relations*

In order to make FrameNet more comprehensive, several frame-to-frame relations have been introduced. Each frame relation in FrameNet is an asymmetric relation between two frames, where one frame (the more abstract) is called the **parent frame** and the other (the less abstract) is called the **child frame**. The set of the most important relations is comprised as follows:

parent frame
child frame

- **Inheritance** – the strongest relation between frames, corresponding to the IS-A relation in ontologies. Each child frame must inherit all properties of its parent frame and share all FEs.
- **Subframe** – some frames are complex in the sense that they refer to sequences of states and transitions, where each of them can separately be described as an individual frame. The separate frames (child frames) are related to the complex frames via the Subframe relation.
- **Perspective_on** – this relation indicates the presence of at least two different points-of-view. For example, a neutral `Commerce_goods_transfer` frame has two points-of-view: `Commerce_buy` and `Commerce_sell`. The neutral frame is usually non-lexical (Ruppenhofer et al., 2006).
- **Using** – the parent frame constitutes the background for its child frame. Not all attributes of the parent frame must be inherited by the child frames. For example, `Volubility` uses the `Communication` frame, since `Volubility` describes a quantification of communication events.

An example of the relations between frames in FrameNet is shown in figure 3.3.

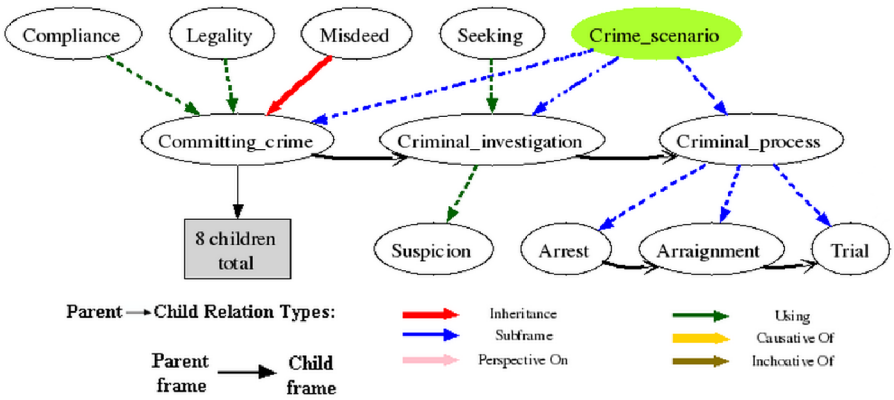


Figure 3.3: FrameNet relation example: `Crime_scenario`.

3.3.4 *Semantic types*

We can say that FrameNet comprises two independent ontologies. The first is a hierarchy of FrameNet frames arranged according to the frame-to-frame relations, especially using the inheritance relation. The second is a hierarchy of **semantic types** (e.g. Sentient for the Cognizer FE), which are connected with some general frame elements.

semantic type

The general use of semantic types in the FrameNet project is to record information that is not representable in frames, nor in the frame element hierarchies. In specific, semantic types are mainly employed for the following functions: indicating the basic types of fillers of frame elements, marking non-lexical types of frames, and recording important semantic differences between lexical units belonging to the same frame.

Each frame element may be connected with one or more semantic types. Unfortunately, only approximately 20 % of all frame elements (usually very general) are connected with any of them. Moreover, the hierarchy of semantic types does not correspond to any existing ontology (although it reminds the Top Ontology in WordNet).

3.4 FRAMENETS IN OTHER LANGUAGES

As the popularity of the Berkeley FrameNet has grown, many research teams throughout the world began developing FrameNets for other languages. The greatest projects are SALSA (The Saarbrücken Lexical Semantics Acquisition Project) (Burchardt et al., 2006) for German, Spanish FrameNet (Subirats and Petruck, 2003) and Japanese FrameNet (Hirose et al., 2004). Other languages for which a FrameNet is being built comprise Chinese, Italian, French, Romanian, Swedish, Brasilian, etc.

3.4.1 *SALSA*

The aim of the SALSA project is to create a German lexical resource annotated by the semantic information following the frame semantics theoretical background, and to investigate methods for its utilization. The SALSA team has chosen a corpus-based approach, which means to extend an existing German treebank TIGER (Brants et al., 2002) by flat trees representing the frame semantic information.

An example of a simple annotation instance is shown in figure 3.4. The root node of a single FrameNet frame is labeled with a frame name, the edges are labeled with frame element names. The verb 'gelten' introduces the frame `Categorization`, the noun phrase 'IBM und Siemens' is annotated as Item and 'als Schimpfworte' as Category.

The annotation process in the SALSA project is fully manual. Two annotators annotate each predicate independently, and in case

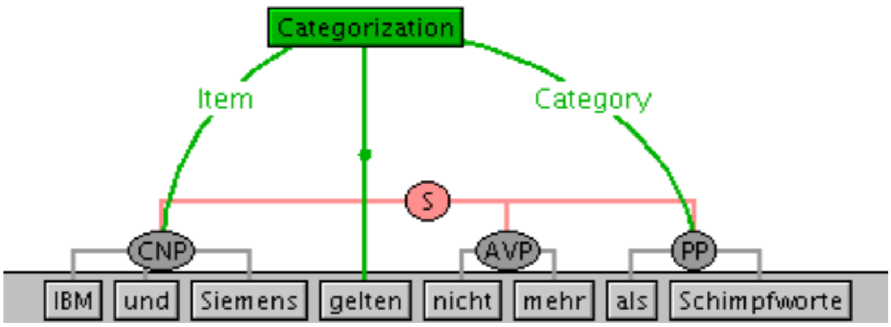


Figure 3.4: SALSA annotation example: `Categorization`.

of disagreement, a third one resolves the collision. Cross-lingual divergences are solved both by adding new frames and new frame elements into the frames.

The current SALSA release 2.0 primarily annotates verbal and noun predicates. The total size of the annotation is currently roughly 20,000 verbal instances and more than 17,000 nominal instances.

3.4.2 *Spanish FrameNet*

Spanish FrameNet is a research project creating an on-line lexical resource for Spanish, based on the frame semantics, and supported by corpus evidence. The aim of the project is to document the range of semantic and syntactic combinatory possibilities (valences) of each word in each of its sense through human-approved and automatic annotated example sentences. The annotated sentences are building blocks of the database. The sentences are marked up in XML and form the basis of the lexical entries. This format supports searching by lemma, frame, frame element, and combinations of these. The Spanish FrameNet Corpus includes both New World and European Spanish, and is composed of texts of different genres, primarily newspapers, newswire texts, book reviews, and humanities essays.

The database acts both as a dictionary and a thesaurus. The dictionary features include definitions and tables showing how frame elements are syntactically expressed in sentences. From the thesaurus perspective, words are linked to the semantic frames in which they participate. The frames, in turn, are linked to word lists and to related frames. The syntactic and semantic annotation is carried out by using the system developed at Berkeley within the FrameNet Project. Each Spanish FrameNet entry provides links to other lexical resources, including Spanish WordNet synsets. The first release of the lexicon is available to the public and contains more than 1,000 lexical items (verbs, predicative nouns, adjectives and adverbs), representing a wide range of semantic domains connected to 325 frames in more than 10,000 sentences.

3.5 BRIEF

An electronic lexicon of Czech verb valencies BRIEF (Pala and Ševeček, 1997) was developed at the Faculty of Informatics, Masaryk University in 1997. It contains about 15,000 Czech verb lemmata and nearly 50,000 valency frames. The lexicon has been built from several printed dictionaries of Czech as well as using an electronic lexicon of Czech stems (Osolsobě and Pala, 1993).

For each verb entry, BRIEF contains a list of frames separated by a comma. The frame is described as a sequence of elements separated by a dash, where each element is represented as a sequence of attribute-value pairs. The attributes are denoted with lower-case letters, the values either as capital letters or as strings delimited by braces. BRIEF was probably the first lexicon of verb valencies for Czech, which served as the cornerstone of its successor, VerbaLex.

3.6 VALLEX

Valency lexicon of Czech verbs, Vallex (Žabokrtský, 2005), has been developing at the Institute of Formal and Applied Linguistics at Charles University in Prague since 2001. Vallex uses the **functional generative description** (Sgall et al., 1986) as its theoretical background and is closely related to the Prague Dependency Treebank (Hajič, 2005). The main goal of the project is to describe valencies of Czech verbs on both the syntactic and semantic layer. The first version of the lexicon, Vallex 1.0, was published in 2003 and contained approximately 1,400 verb entries. The current version, Vallex 2.6, contains almost 4,300 entries.

functional generative description

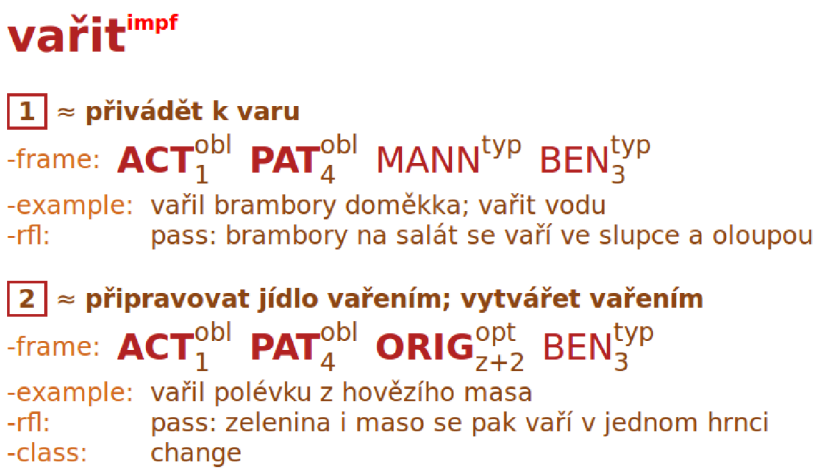


Figure 3.5: Example of the Vallex frame for 'vařit'.

The consistency of verbs in Vallex is improved by grouping them into verb classes of semantically similar verbs. The Vallex classes are based on the so-called alternation model (Lopatková et al., 2006), and currently consists of 22 classes.

The key information on the valency structure of a verb is encoded in the form of valency frames. The valency frames are stored as a sequence of slots, where each slot represents one valency complementation, and consists of its type, morphemic realization and its obligatoriness. The position of a verb is not marked in the Vallex frame. An example of the frame for verb 'vařit' (to boil or cook) is shown in Figure 3.5.

3.7 VERBALEX

VerbaLex (Hlaváčková and Horák, 2006) is an electronic database of verb valency frames in Czech, which has been developed in the Centre for Natural Language Processing at the Faculty of Informatics, Masaryk University. Basic units (entries) in VerbaLex consist of verb lemmata grouped into synsets together with their sense numbers in the standard WordNet notation. Verb valencies are realized on two levels – deep valency level, which corresponds to a semantic role (semantic values of the verb arguments), and surface level, reflecting the information about syntactic and morphological valencies. The current version of VerbaLex contains more than 6,000 synsets, more than 21,000 verb senses, and approximately 10,500 verb lemmata in 19,500 valency frames.

A valency frame represents verb valencies on both syntactic and semantic level. In the centre of the frame, there is a mark representing the verb position, surrounded by left-hand and right-hand arguments in the canonical word-order. The type of the valency relation for each constituent element is marked as obligatory or optional. Semantic information about the verbal complement is represented by two-level semantic roles.

The first level is represented with main labels primarily based on the EuroWordNet (Vossen and Hirst, 1998) first-order and second-order top ontology entities arranged in a hierarchical structure. The inventory of these labels is closed and currently contains 33 items (concepts). On the second level, there is a collection of selected lexical units (literals) from the set of EuroWordNet and BalkaNet base concepts (Pala and Smrž, 2004) with their respective sense numbers. The list of second level semantic roles is open and currently contains about 1,200 literals.

The valency frames also contain other additional information about verbs. By combining all this information we obtain a so called Complex Valency Frame (CVF). The additional information from CVF includes:

- definition of verb meaning
- verb's ability to create a passive form
- number of meanings for homonymous verbs
- semantic class which a verb belongs to
- aspect (perfective or imperfective)
- example of verb use
- types of reflexivity for reflexive verbs

An example of the VerbaLex frame for 'vařit' (to boil or cook) is shown in figure 3.6.

vařit se$_2^{impf}$ vařit$_2^{impf}$

definition: *tepelně upravovat jídlo*
passive: yes (vařit); no (vařit se)
English equivalent: ENG20-00317835-v [+] Show more PWN information

1 vařit$_2$ ≈
-frame: **AG**$_{kdo1}^{obl}$ **VERB**^{obl} **SUBS**$_{co4}^{obl}$
-example: *vaří guláš* ***(impf)***
-synonym: vařit se$_2$
-use: prim
-reflexivity: no
2 vařit se$_2$ ≈
-frame: **SUBS**$_{co4}^{obl}$ **VERB**^{obl}
-example: *brambory se vaří* ***(impf)***
-synonym: vařit$_2$
-use: prim
-reflexivity: refl

Figure 3.6: Example of the VerbaLex frame for 'vařit'.

Part II

PROBABILISTIC SEMANTIC FRAMES

All models are wrong, but some are useful.

(Box, 1987)

4 PROBABILISTIC GRAPHICAL MODELS

Probabilistic models are very powerful tools for describing real world problems. Their representation using complex mathematical equations, however, is not convenient for readers. We can therefore find it very useful to augment the algebraic description with diagrammatic representations of probability distributions. These graphical representations of probabilistic models are usually called **probabilistic graphical models** (Koller and Friedman, 2009). Many people use them for their ability to provide a simple way to visualize the structure of a probabilistic model. Insights into the properties of the model, for example the conditional independence or relations between random variables, can easily be obtained by inspection of the graph. Moreover, computations required to perform inference and learning in such visualized models can be expressed in terms of simple graphical manipulations.

probabilistic graphical model

A Probabilistic Graphical Model (PGM) comprises **nodes**, which represent individual random variables, and **edges** that express probabilistic relations between these variables. The graph captures the way in which the joint distribution over all the random variables can be decomposed into a product of factors. These factors then depend only on a subset of adjacent random variables. We can distinguish two basic types of PGMs – **Bayesian networks** that use directed edges, and **Markov random fields** in which the links between nodes do not carry arrows and have no directional significance. Directed graphs are useful for expressing causal relationships between random variables, whereas undirected graphs capture symmetric dependencies between random variables. Since the proposed model of probabilistic semantic frames is based on causal relationships, we will only focus on Bayesian networks.

graph node

graph edge

Bayesian network

Markov random field

4.1 BAYESIAN NETWORKS

Bayesian networks are probabilistic graphical models that use a graph-based representation for compactly encoding a complex distribution over a high-dimensional space. In order to motivate the use of directed graphs to describe probability distributions, consider the example in figure 4.1.

The figure shows a toy example of a Bayesian network describing relations in a subset of random variables that have direct or indirect impact on the event of being at work on time. The choice of the vehicle as well as whether it rains depends directly on the current weather.

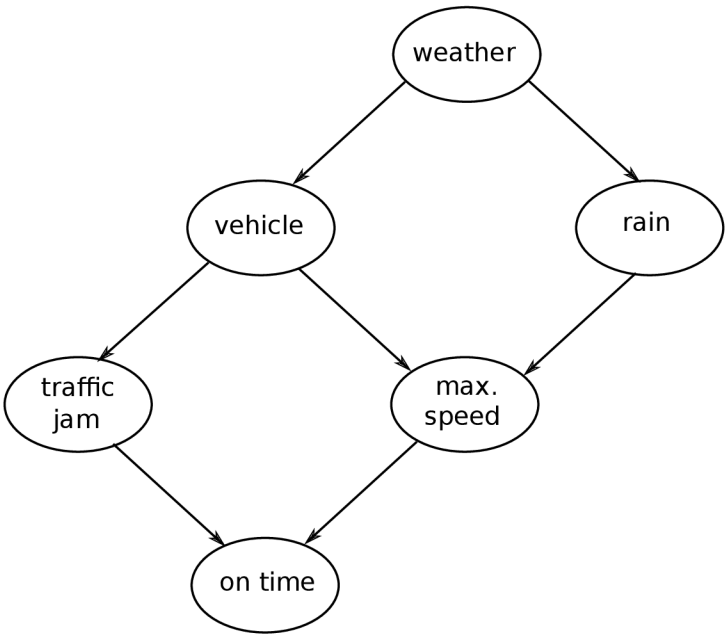


Figure 4.1: Bayesian Network.

Being stuck in a traffic jam depends on the chosen vehicle, maximum speed depends on the chosen vehicle and on rain, and finally, being on time at work depends on the traffic jam and maximum speed.

conditional independence

The graph is, on one hand, a compact representation of the set of **conditional independences** that hold in the distribution, and, on the other hand, a skeleton for factorizing the probability distribution. A random variable X is independent of Y given Z, denoted $X \perp Y | Z$, if

$$P(X|Y,Z) = P(X|Z). \tag{4.1}$$

There are several conditional independences in figure 4.1, for example:

$$\texttt{vehicle} \perp \texttt{rain} \,|\, \texttt{weather} \tag{4.2}$$

$$\texttt{weather} \perp \texttt{max. speed} \,|\, \texttt{vehicle, rain.} \tag{4.3}$$

factor

The other perspective is that the graph defines a skeleton for representing a high-dimensional distribution. Rather than encoding the probability for every possible assignment of values to all random variables in our model, we can break up the distribution into smaller **factors**. The overall joint distribution is then defined as the product of all the factors. If we label, to be short, each random variable by its first letter, the joint distribution of the model in figure 4.1 is defined as

$$\begin{aligned} P(W,V,R,T,M,O) = \\ P(W)P(V|W)P(R|W)P(T|V)P(M|V,R)P(O|T,M). \end{aligned} \tag{4.4}$$

This parametrization is significantly more compact and tractable. It turns out, moreover, that both these perspectives are, in a way, equivalent. The independence properties are what allows it to be presented in a factorized form. Conversely, a particular factorization guarantees that certain independences hold.

4.1.1 *Model Representation*

The Bayesian network in figure 4.1 was an example of a very small network with discrete random variables (apart from `max. speed` which can be continuous). Real-world describing networks, though, are much bigger with a lot of continuous variables, and we thus need a compact notation to represent them. The solution is known as **plate notation**.

plate notation

As an illustration of the plate notation, we consider a model describing the distribution of people's heights in the population. It is well known that the heights are distributed according to the **normal (Gaussian) distribution**, governed with two parameters – μ, called **mean**, and σ^2, called **variance**:

normal distribution

mean

variance

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}. \tag{4.5}$$

The square root of the variance, denoted as σ, is called the **standard deviation**, and the reciprocal of the variance, denoted as $\beta = 1/\sigma^2$, is called the **precision**. Figure 4.2[1] shows the plot of the normal distribution. Given a sample of people's heights from the population

standard deviation

precision

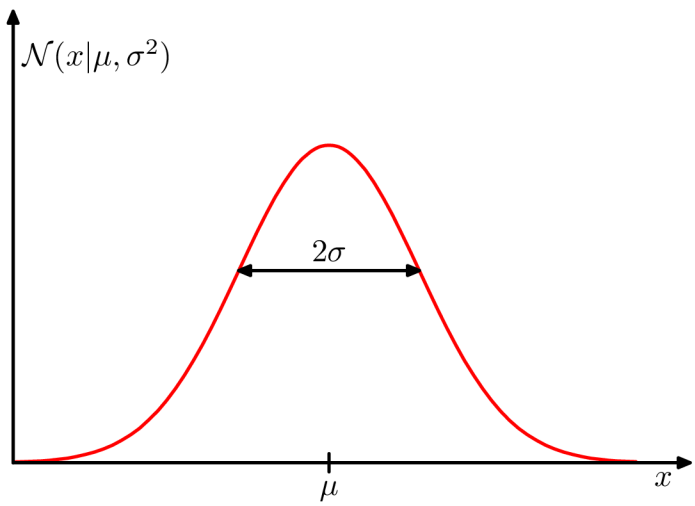


Figure 4.2: Plot of the univariate Gaussian showing the mean μ and the standard deviation σ.

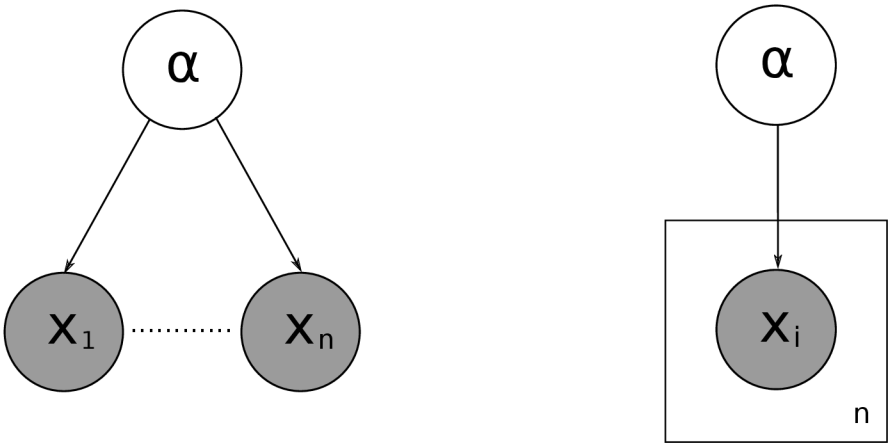
1 The figure is adopted from Bishop (2007).

$\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ distributed according to $\mathbf{x} \sim \mathcal{N}(\mu, \sigma^2)$, our goal might be to find the parameters $\boldsymbol{\alpha} = (\mu, \sigma^2)$ such that

$$\boldsymbol{\alpha} = \arg\max_{\bar{\boldsymbol{\alpha}}} P(\mathbf{x}|\bar{\boldsymbol{\alpha}}), \tag{4.6}$$

which is called the Maximum-Likelihood Estimation (MLE). Even though this probabilistic model is very simple, it might be useful to represent it using a Bayesian network.

The probabilistic graphical model is shown in figure 4.3a. The people's individual heights are represented as random variables $x_1, x_2, \ldots, x_n$, which depend on the mean and the variance of normal distribution, represented together by one random variable $\boldsymbol{\alpha}$. This joint notation of the parameters is only used to simplify the figure.



(a) Expanded notation. (b) Plate notation.

Figure 4.3: Probabilistic graphical model for the distribution of people's heights in the population.

When we apply a graphical model to a problem in machine learning or pattern recognition, we typically set some of the random variables to specific observed values, while the other variables remain non-observed and are supposed to be learned. One can see that the nodes of variables $x_1, x_2, \ldots, x_n$ are shaded to indicate that they are **observed variables**. It means that they have been set according to their values in the training data. Conversely, the random variable without shading $\boldsymbol{\alpha}$ is a **latent variable**, also known as a **hidden variable**, whose value is under investigation. This is a standard notation that will be followed throughout the rest of this thesis.

observed variable

latent variable
hidden variable

When we deal with more complex graphical models with multiple random variables of the same type, it is inconvenient to write out all of them explicitly as in 4.3a. Instead of drawing each variable individually, a rectangle can be used to group them together into a subgraph. The plate notation for our example is shown in 4.3b. The number n, placed in the corner, represents the number of repetitions of the subgraph in the plate. Any edges that cross a plate boundary

are replicated once for each subgraph repetition as well. In our example, the replicated variables are indexed with i.

Both the toy probabilistic graphical models in figures 4.1 and 4.3 satisfy an important property – there are no cycles. In fact, Bayesian networks do not allow any type of cycles in their structure under any circumstances. This is an equivalent to the statement that there exists an ordering of the nodes such that there are no links that go from any node to any lower numbered node. Such directed graphs without cycles are called **directed acyclic graphs**.

directed acyclic graph

Formally, directed graph is a data structure $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ consisting of a set of nodes $\mathcal{X}$ and a set of edges $\mathcal{E} : \mathcal{X} \to \mathcal{X}$. We say that $X_1, X_2, \ldots, X_k$ for $X_i \in \mathcal{X}$ is a path in the graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ if, for every $i = 1, 2, \ldots, k-1$, we have $(X_i \to X_{i+1}) \in \mathcal{E}$. Next, we define a **cycle** in $\mathcal{G}$ as a path $X_1, X_2, \ldots, X_k$ where $X_1 = X_k$. A graph is **acyclic** if it contains no cycles.

cycle
acyclic

Finally, the Bayesian network is defined as a directed acyclic graph $\mathcal{G} = (\mathcal{O} \cup \mathcal{H}, \mathcal{E})$, where $\mathcal{O}$ is a set of observed variables, $\mathcal{H}$ is a set of hidden variables and $\mathcal{E} : \mathcal{O} \cup \mathcal{H} \to \mathcal{O} \cup \mathcal{H}$ is a set of oriented edges that satisfy the acyclic condition.

4.2 GENERATIVE VS. DISCRIMINATIVE MODELS

The goal of exploiting probabilistic graphical models in machine learning applications is typically to predict the values of latent variables based on a model and training data. This process is called **statistical inference** or **statistical induction**. As we saw in our example in figure 4.3, the goal was to predict the mean and the variance of people's heights in the population. The process is usually broken into two separate stages: the inference stage, where we learn the probability distribution of a hidden variable z given training data $\mathbf{x}$, and the decision stage in which we use the probability distribution $P(z|\mathbf{x})$ to make an optimal choice of the value of hidden variable z. When the hidden variable z is discrete, the machine learning task is typically called **classification**, in case of a continuous variable, it is called **regression**. Since the problems in natural language processing are mostly of a discrete nature, we will focus only on the classification task.

statistical inference
statistical induction
classification
regression

In general, we can identify two distinct models to estimate the posterior probability over the values of hidden variable z – **generative** model and **discriminative** model. Let us summarize the main differences.

generative model
discriminative model

The generative model estimates the probability distributions $P(z)$ and $P(\mathbf{x}|z)$, and combines them using the Bayes' theorem into the posterior probability $P(z|\mathbf{x})$, whereas the discriminative model estimates the posterior probability directly:

- **Generative models** for the classification tasks solve the inference problem by determining the probability densities for each class z_c of the discrete random variable z separately. Then it is possible to use the Bayes' theorem in the form

$$P(z_c|\mathbf{x}) = \frac{P(\mathbf{x}|z_c)P(z_c)}{P(\mathbf{x})} \tag{4.7}$$

to find the posterior probabilities $P(z_c|\mathbf{x})$, where the denominator $P(\mathbf{x})$ can be found as the sum of the quantities appearing in the numerator:

$$P(z_c|\mathbf{x}) = \frac{P(\mathbf{x}|z_c)P(z_c)}{\sum_c P(\mathbf{x}|z_c)P(z_c)}. \tag{4.8}$$

The posterior probabilities in generative models can equivalently be determined by modeling the joint distribution $P(z_c, \mathbf{x})$, which is equal to $P(z_c)P(\mathbf{x}|z_c)$. Similarly to the previous case, the normalization is required:

$$P(z_c|\mathbf{x}) = \frac{P(z_c, \mathbf{x})}{\sum_c P(z_c, \mathbf{x})}. \tag{4.9}$$

Such models are called generative because they model inputs as well as outputs, thus, it is possible to generate synthetic data points in the input space.

- **Discriminative models** solve the inference problem by determining the posterior probability $P(z_c|\mathbf{x})$ directly. That is why it is impossible to generate synthetic data. They can only solve the assignment of the most probable class to the input data.

naïve Bayes classifier

logistic regression

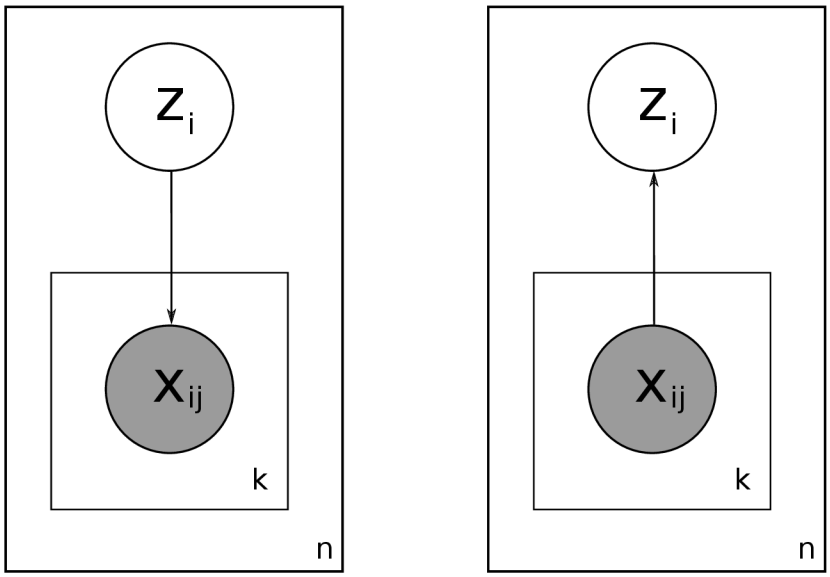
Both approaches can easily be exemplified in the classification task, where the generative model is represented by the **naïve Bayes** classifier and the discriminative model by the **logistic regression** classifier. Their graphical models are shown in figure 4.4.

Naïve Bayes classifier, shown in figure 4.4a, is a simple probabilistic classifier with a strong independence assumption. The model assumes that the attributes $x_{i1}, x_{i2}, \ldots, x_{ik}$ of data points $x_1, x_2, \ldots, x_n$ are conditionally independent of one another given the class variable z_i. This significantly reduces complexity of the model and requirements for the size of training data n. The algorithm solves the problem of classification by applying the Bayes' theorem

$$P(z_c|\mathbf{x_i}) = \frac{P(\mathbf{x_i}|z_c)P(z_c)}{\sum_c P(\mathbf{x_i}|z_c)P(z_c)}, \tag{4.10}$$

where $P(\mathbf{x_i}|z_c)$ is estimated using the conditional independence assumption

$$P(\mathbf{x_i}|z_c) = \prod_{j=1}^{k} P(x_{ij}|z_c). \tag{4.11}$$



(a) Naïve Bayes classifier. (b) Logistic regression classifier.

Figure 4.4: Generative and discriminative types of classifiers.

It can be seen that the naïve Bayes classifier model describes how to generate data points based on known classes.

Contrarily, the logistic regression classifier (figure 4.4b) directly describes how to predict the classes based on the attributes of data points. In order to simplify the notation, we consider the case where z is a boolean variable (i.e. $z = 0$ or $z = 1$). The model assumes a parametric form for the distribution given by

$$P(z = 0|\mathbf{x}_i) = \frac{1}{1 + \exp(w_0 + \sum_{j=1}^{k} w_j x_{ij})} \tag{4.12}$$

and

$$P(z = 1|\mathbf{x}_i) = \frac{\exp(w_0 + \sum_{j=1}^{k} w_j x_{ij})}{1 + \exp(w_0 + \sum_{j=1}^{k} w_j x_{ij})} \tag{4.13}$$

where $w_0, w_1, \ldots, w_k$ are parameters of the model which are supposed to be learned. The model makes use of the logistic function, which transforms any real number into the interval $\langle 0, 1 \rangle$. The logistic function curve, defined as

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.14}$$

is shown in figure 4.5.

The examples of generative and discriminative classifiers illustrate their most important properties. On one hand, the generative approach is in general more demanding on the size of training data because it is required to find the joint distribution over both

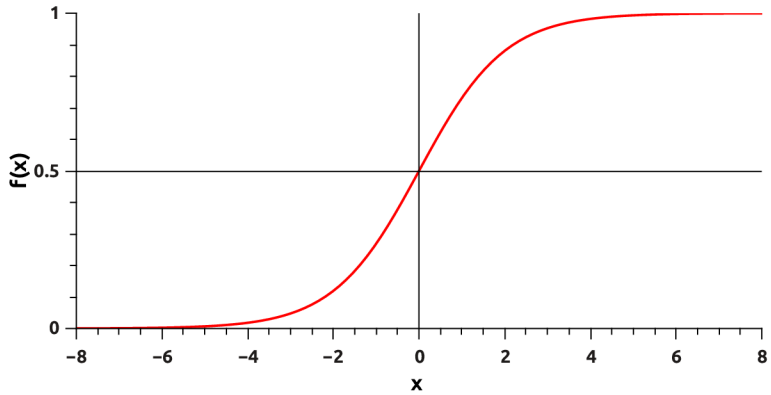


Figure 4.5: Logistic function.

training data and the classes. That is why the naïve Bayes is more popular than a "non-naïve" generative approach that does not require the conditional independence between attributes. On the other hand generative approaches are usually easier to design and implement, and most importantly, generative models usually do not require hand-labeled data. The requirement of a plentiful amount of labeled training data is an important disadvantage of discriminative techniques.

Since the goal of this thesis is to design an unsupervised model trained on a huge amount of non-labeled data which is easily acquired, the generative models will be preferred and used in the proposed probabilistic semantic frames models.

4.3 STATISTICAL INFERENCE IN BAYESIAN NETWORKS

statistical inference

The process of computing posterior distributions over hidden variables and finding their most likely values in probabilistic models is usually called **statistical inference**. The inference in small networks with tens or hundreds of nodes can be done using exact inference techniques such as variable elimination or belief propagation (Kim and Pearl, 1983). For larger networks, however, the exact inference is too slow, because it is an NP-hard problem (Cooper, 1990). Since our semantic frame models will be comprised of millions of nodes, we will only focus on fast, approximate inference techniques.

variational inference

Markov chain Monte Carlo

The first of the two approaches for approximate inference in Bayesian networks that will be discussed in this chapter is called **variational inference**. It is a deterministic scheme based on analytical approximation to the posterior distribution. The second approach relies on stochastic approximations. We will only introduce a small subset of stochastic sampling algorithms known as **Markov Chain Monte Carlo**.

4.3.1 *Variational Inference*

Variational inference has its origin in the work on the **calculus of variations**, which is a field of mathematical analysis that deals with maximizing or minimizing **functionals**. Functional is a function $f : \mathbf{V} \to \mathbb{R}$ that takes a function from a space of functions $\mathbf{V}$ (or a vector from some vector space) as input and returns a scalar number. While in the standard calculus a derivative describes how the output value varies as we make infinitesimal changes to the input variable, in the calculus of variation, we analyze how the value of functional changes in response to infinitesimal changes of the input function. For details and definitions, see e.g. Hazewinkel (2001).

calculus of variations
functional

The idea of approximation using the variational inference is that we restrict the range of functions describing the true densities of a hidden variable by considering only tractable functions. The optimization task lies in searching for functions that are the closest to the true posteriors.

Suppose we have a Bayesian network model where all latent variables as well as parameters are denoted as $\mathbf{Z} = \{z_1, z_2, \ldots, z_m\}$, and the set of all observed variables is $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$. The probabilistic model specifies the joint distribution $P(\mathbf{Z}, \mathbf{X})$, and our goal is to find an approximation for the posterior distribution $P(\mathbf{Z}|\mathbf{X})$, which is denoted $Q(\mathbf{Z})$ and satisfies

$$Q(\mathbf{Z}) \approx P(\mathbf{Z}|\mathbf{X}). \tag{4.15}$$

The lack of similarity between the approximation $Q(\mathbf{Z})$ and the true posterior $P(\mathbf{Z}|\mathbf{X})$ is measured by a dissimilarity function $d(Q, P)$. The most common choice for the dissimilarity function is the **Kullback-Leibler divergence** (Kullback and Leibler, 1951), which is defined as

Kullback-Leibler divergence

$$KL(Q||P) = \sum_{Z} Q(\mathbf{Z}) \ln \frac{Q(\mathbf{Z})}{P(\mathbf{Z}|\mathbf{X})}. \tag{4.16}$$

It is a non-symmetric measure of the difference between two probability distribution Q and P. More specifically, $KL(Q||P)$ measures the information loss when Q is used to approximate P, i.e., it measures the expected number of extra bits required to code samples from P when using a code based on Q, rather than using a code based on P (Burnham and Anderson, 2002).

The KL-Divergence can be decomposed as is shown in following formulas:

$$KL(Q||P) = \sum_{Z} Q(\mathbf{Z}) \ln \frac{Q(\mathbf{Z})}{P(\mathbf{Z}, \mathbf{X})} + \ln P(\mathbf{X}), \tag{4.17}$$

$$\ln P(\mathbf{X}) = KL(Q||P) + \mathcal{L}(Q), \tag{4.18}$$

where the term $\mathcal{L}(Q)$ is known as the negative variational free energy and is defined as

$$\mathcal{L}(Q) = -\sum_{Z} Q(\mathbf{Z}) \ln \frac{Q(\mathbf{Z})}{P(\mathbf{Z}, \mathbf{X})}. \tag{4.19}$$

Since the KL-divergence is always non-negative and the log evidence $\ln P(\mathbf{X})$ is fixed with respect to Q, we can observe that maximizing $\mathcal{L}(Q)$ minimizes the dissimilarity $KL(Q||P)$. If we allow any possible choice for $Q(\mathbf{Z})$, then the maximum of the lower bound $\mathcal{L}(Q)$ occurs when the KL-divergence vanishes, which happens when $Q(\mathbf{Z})$ equals the true posterior distribution $P(Z|X)$. However, the true posterior is usually intractable, so we are doomed to use a simpler form of the distribution Q.

Mean-Field approximation

The most common type of approximation for variational inference is known as **Mean-Field approximation** (Parisi, 1998). The method supposes decomposition of the distribution over $\mathbf{Z}$ into disjoint groups that we denote $\mathbf{Z}_i$, where $i = 1, 2, \ldots, F$. We then assume that the distribution Q factorizes with respect to these groups:

$$Q(\mathbf{Z}) = \prod_{i=1}^{F} Q_i(\mathbf{Z}_i), \tag{4.20}$$

where there is no restriction on the functional forms of the individual factors $Q_i(\mathbf{Z}_i)$.

Variational methods have their strengths and weaknesses – their strongest advantage is that they are deterministic, so they always find the same solution after a specific number of iterations. They are quite fast, but they do not guarantee the perfect accuracy if we use a non-optimal approximation Q. Moreover, it is not always easy to design and implement the inference algorithm based on variational inference if the models are complex. These are the reasons why we will prefer sampling methods, which guarantee a specific accuracy after large enough number of iterations. They will be described in the subsequent section.

4.3.2 *Sampling*

Monte Carlo

The second large group of approximate inference techniques is based on numerical sampling, and is known as the collection of **Monte Carlo** techniques. It was named after a well known casino in Monte Carlo, where an uncle of the inventor of these techniques, Stanisław Ulman, often gambled (Eckhardt, 1987). The heart of modern Monte Carlo simulations lies in selecting a statistical sample to approximate a hard combinatorial problem.

Sampling and Monte Carlo methods have many applications in physics and applied mathematics (Newman and Barkema, 1999), but

we will only focus on evaluating expectations over hidden variables in Bayesian networks in order to make predictions. A fundamental problem in statistical inference involves finding the expectation of a function $f(z)$ with respect to the probability distribution $P(z)$. Thus, in case of k discrete values of the hidden variable z, the goal is to find the expectation

$$\mathbb{E}[f] = \sum_{i=1}^{k} f(z_i)P(z_i). \tag{4.21}$$

We shall suppose that such expectations are too complex to be evaluated exactly using analytical techniques, so some sort of approximation is required. The general idea is to obtain a set of samples $z^{(1)}, z^{(2)}, \ldots, z^{(L)}$, drawn independently from the distribution $P(z)$. This allows the expectation to be approximated by a finite sum

$$\hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(z^{(l)}). \tag{4.22}$$

Assuming that it is possible to sample from the conditional distribution at each node of the probabilistic graphical model, and that the graph contains no observed variables, sampling from the joint distribution of the whole model is straightforward. The joint distribution is defined as

$$P(\mathbf{Z}) = \prod_{i=1}^{m} P(z_i|\text{parents}_i), \tag{4.23}$$

where parents_i denotes the set of variables associated with the parents of node i. To obtain the sample from the joint distribution, we can pass through the graph in the direction of edges and sample from the conditional distribution $P(z_i|\text{parents}_i)$. It is always possible because at each step, all parent nodes will have been instantiated. After we pass through the whole graph, we will obtain a sample from the joint distribution.

For the Bayesian networks with discrete hidden variables and some observed variables, we can extend the procedure above. At each step, when a sampled value is obtained for a variable z_i whose value is observed, the sample is compared with the observed variable. If the value of the observed variable is equal to the sampled value, we can continue with sampling subsequent variables. However, if the observed value and the sampled value disagree, the values of previously sampled variables have to be discarded, and the process of sampling the joint distribution has to start from the first node in the graph.

This algorithm samples correctly from the true joint posterior, however, the probability of accepting a sample from the posterior rapidly decreases with a growing number of observed variables

and the number of their states. Moreover, this approach is not applicable to models with continuous observed variables, and so it is rarely used in practice. There are several Monte Carlo methods that avoid discarding sampling as much as possible. The most common techniques are described below.

4.3.2.1 *Rejection sampling*

rejection sampling

The idea of the **rejection sampling**, see Bishop (2007) for instance, is based on transforming the task to sampling from a simpler distribution. Suppose we want to sample from a distribution $P(z)$, which is not known. Instead, we can sample from another easy-to-sample distribution $Q(z)$, called the **proposal distribution** that satisfies $P(z) \leqslant kQ(z)$.

proposal distribution

Each step of the rejection sampler involves generating two random numbers. First, we generate a random number $z^{(i)}$ from the distribution $Q(z)$. Next, we generate a number u from the uniform distribution over $\langle 0, 1 \rangle$. If

$$u < \frac{P(z^{(i)})}{kQ(z^{(i)})} \tag{4.24}$$

then the sample is accepted, otherwise $z^{(i)}$ is rejected. The complete scenario is described in algorithm 4.1.

Algorithm 4.1 Rejection sampling algorithm for N iterations.

```
i ← 0
while i < N do
    Sample z^(i) ~ Q(z)
    Sample u ~ Uniform(0, 1)
    if u < P(z^(i)) / (kQ(z^(i))) then
        Accept z^(i)
        i ← i + 1
    else
        Reject
    end if
end while
```

It can easily be shown that the accepted samples $z^{(i)}$ are sampled with probability $P(z)$ (Robert and Casella, 2005). The most limiting property of the rejection sampling is that it is not always possible to find a reasonable constant k. If k is too large, the acceptance probability is too small, which makes the method impractical.

4.3.2.2 *Markov Chain Monte Carlo*

It is straightforward that the rejection sampling is inefficient for high-dimensional random spaces, because the rejections come too often.

We therefore turn in this section to a very general and powerful framework called Markov Chain Monte Carlo (MCMC), which allows sampling from high-dimensional spaces easily. The exploration of the MCMC methods will start with the **Metropolis-Hastings** algorithm (Metropolis et al., 1953; Hastings, 1970), which is a very popular method for sampling. We will see that the most practical MCMC algorithm, Gibbs sampling, which will be used mostly in the rest of the thesis, can be interpreted as a special case of this algorithm.

Metropolis-Hasting algorithm

All MCMC algorithms are built on the idea of the **Markov chain**. The Markov chain is a sequence of random variables $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$, where the probability of the value of variable $x^{(t)}$ depends only on the value of variable $x^{(t-1)}$, i. e.:

Markov chain

$$P(x^{(t+1)} = s_j | x^{(1)} = s_k, \ldots, x^{(t)} = s_i) = P(x^{(t+1)} = s_j | x^{(t)}) = s_i). \tag{4.25}$$

This property is known as the **Markov property**.

Markov property

More precisely, we can define the Markov chain in terms of a graph of states $Val(\mathcal{X})$ and a transition model $\mathcal{T}$ that defines, for every state $s_i \in \mathcal{X}$ a next-state distribution over $\mathcal{X}$. The probability $P(i \to j)$ of going from state s_i to state s_j in one step is referred to as the **transition probability**:

transition probability

$$P(i \to j) = P(x^{(t+1)} = s_j | x^{(t)} = s_i). \tag{4.26}$$

This probability is applied whenever the chain is in state s_i. Let

$$\pi_j(t) = P(x^{(t)} = s_j) \tag{4.27}$$

denote the probability that the chain is in state s_j at time t, and let $\boldsymbol{\pi}(t)$ denote the vector of probabilities for all the states at time t. The probability that the chain is in state s_i at time $t+1$ is given by the **Chapman-Kolomogrov equation**:

Chapman-Kolomogrov equation

$$\pi_i(t+1) = P(x^{(t+1)} = s_i) = \sum_k P(k \to i)\pi_k(t). \tag{4.28}$$

We can compactly write the Chapman-Kolomogrov equation for all random variables in so called **transition matrix** P, where the element at position (i, j) is equal to $P(i \to j)$. The Chapman-Kolomogrov equation then becomes

transition matrix

$$\boldsymbol{\pi}(t+1) = \boldsymbol{\pi}(t)P. \tag{4.29}$$

Finally, a Markov chain is said to be **irreducible** if there exists a positive probability of visiting all states in finite time, starting from any state of the chain. Likewise, a chain is said to be **aperiodic** if the greatest common divisor of the lengths of its cycles is equal to one. Being irreducible and aperiodic, every Markov chain can

irreducibility

aperiodicity

stationary distribution

reach a **stationary distribution** $\boldsymbol{\pi}^*$, where the vector of transition probabilities is independent of the initial conditions:

$$\boldsymbol{\pi}^* = \boldsymbol{\pi}^* P. \tag{4.30}$$

reversibility condition

A sufficient condition for a unique stationary distribution is the **reversibility condition**

$$P(j \to k)\pi_j^* = P(k \to j)\pi_k^*, \tag{4.31}$$

because

$$(\boldsymbol{\pi}P)_j = \sum_i \pi_i P(i \to j) = \sum_i \pi_j P(j \to i) = \pi_j \sum_i P(j \to i) = \pi_j. \tag{4.32}$$

4.3.2.3 *Metropolis-Hastings sampling*

Let us first introduce the basic Metropolis algorithm (Metropolis and Ulam, 1949; Metropolis et al., 1953). As in the rejection sampling, we again sample from a proposal distribution. This time, however, we maintain a record of the last state $x^{(t-1)}$. The proposal distribution $Q(x^*|x^{(t-1)})$ then depends on this state, so that the sequence forms a Markov chain. Suppose our goal is to draw samples from $P(x)$, where

$$P(x) = kf(x), \tag{4.33}$$

and the normalizing constant k is not known or very difficult to compute. The proposal distribution is chosen to be sufficiently simple so that it is straightforward to draw samples from it directly. Moreover, it must be a symmetric distribution. It means that $Q(x^{(1)}|x^{(2)}) = Q(x^{(2)}|x^{(1)})$ for all values of $x^{(1)}$ and $x^{(2)}$.

The algorithm starts with the initial value $x^{(0)}$ satisfying $f(x^{(0)}) > 0$. Using the current value $x^{(i)}$ it samples a candidate value x^* from the proposal distribution $Q(x^*|x^{(i)})$. And finally, given the candidate x^*, it calculates the ratio of the density at the candidate x^* and the current state $x^{(i)}$:

$$\alpha = \frac{P(x^*)}{P(x^{(i)})} = \frac{kf(x^*)}{kf(x^{(i)})}. \tag{4.34}$$

Notice that because we are considering the ratio under two different values, the normalizing constant k cancels out. If the ratio is greater or equal to one ($\alpha \geqslant 1$), we accept the candidate point. If the ratio is lower than one ($\alpha < 1$), we accept it with probability α, or else reject it. The procedure is summarized in algorithm 4.2.

Hastings (1970) generalized the Metropolis algorithm by using an arbitrary proposal distribution $Q(x^*|x^{(i)})$. It can be achieved by replacing the acceptance probability defined at line 6 of the algorithm 4.2 with a new definition given by

$$u < \min\left(1, \frac{f(x^*)Q(x^*|x^{(i)})}{f(x^{(i)})Q(x^{(i)}|x^*)}\right). \tag{4.35}$$

Algorithm 4.2 Metropolis sampling algorithm for N iterations.

```
Initialize x^(0)
i ← 0
while i < N do
    Sample x* ~ Q(x*|x^(i))
    Sample u ~ Uniform(0, 1)
    if u < min(1, f(x*)/f(x^(i))) then
        Accept x^(i)
        i ← i + 1
    else
        Reject
    end if
end while
```

To show that the Metropolis-Hastings algorithm generates samples from the expected density $P(x)$, it is sufficient to show that the generated Markov chain satisfies the reversibility condition from equation 4.31 (Walsh, 2004). In the algorithm, we sample from $Q(x^*|x^{(i)}) = P(x^{(i)} \to x^*|Q) = Q(x^{(i)} \to x^*)$ and then accept with probability

$$\alpha(x^{(i)} \to x^*) = \min\left(1, \frac{f(x^*)Q(x^*|x^{(i)})}{f(x^{(i)})Q(x^{(i)}|x^*)}\right), \tag{4.36}$$

so the transition probability is $P(x^{(i)} \to x^*) = Q(x^*|x^{(i)})\alpha(x^{(i)} \to x^*)$. Thus if the Markov chain satisfies

$$Q(x \to y)\alpha(x \to y)P(x) = Q(y \to x)\alpha(y \to x)P(y) \tag{4.37}$$

for all x, y, then the stationary distribution corresponds to draws from the target distribution. To show that the balance equation is satisfied, consider three possible cases for the values of x, y:

1) If $Q(x \to y)P(x) = Q(y \to x)P(y)$

   then $\alpha(x \to y) = \alpha(y \to x) = 1$, which implies
   $P(x \to y)P(x) = Q(x \to y)P(x)$, $P(y \to x)P(y) = Q(y \to x)P(y)$
   and hence
   $P(x \to y)P(x) = P(y \to x)P(x)$.

2) If $Q(x \to y)P(x) > Q(y \to x)P(y)$

   then
   $\alpha(x \to y) = \dfrac{P(y)Q(y \to x)}{P(x)Q(x \to y)}, \qquad \alpha(y \to x) = 1$

and hence

$$\begin{aligned}
P(x \to y)P(x) &= Q(x \to y)\alpha(x \to y)P(x) \\
&= Q(x \to y)\frac{P(y)Q(y \to x)}{P(x)Q(x \to y)}P(x) \\
&= Q(y \to x)P(y) \\
&= Q(y \to x)\alpha(y \to x)P(y) \\
&= P(y \to x)P(y).
\end{aligned}$$

3) If $Q(x \to y)P(x) < Q(y \to x)P(y)$

then
$\alpha(y \to x) = \dfrac{P(x)Q(x \to y)}{P(y)Q(y \to x)}, \qquad \alpha(x \to y) = 1$
and hence

$$\begin{aligned}
P(y \to x)P(y) &= Q(y \to x)\alpha(y \to x)P(y) \\
&= Q(y \to x)\frac{P(x)Q(x \to y)}{P(y)Q(y \to x)}P(y) \\
&= Q(x \to y)P(x) \\
&= Q(x \to y)\alpha(x \to y)P(x) \\
&= P(x \to y)P(x).
\end{aligned}$$

burn-in period

An important issue with MCMC samplers, including the Metropolis-Hastings algorithm, is known as the **burn-in period**. It is the phase at the beginning of sampling until the chain approaches the stationarity. Samples from this phase are usually thrown out. Typically the burn-in period lasts for the first several hundreds of iterations, and then one of the various convergence tests, e.g. (Geyer, 1992; Gelman and Rubin, 1992; Geweke, 1992; Raftery and Lewis, 1992), is used to assess whether the stationarity has indeed been reached.

A poor choice of starting values, however, can greatly increase the number of required iterations. A general suggestion is to start the chain as close to the center of the distribution as possible if the center can be estimated. Moreover, if the distribution $P(x)$ is multimodal, the Markov chain can trap close to one of the modes. This can be partially solved by accepting only every k^{th} sample after the burn-in period. Another solution is to start multiple chains with different initial values or use **simulated annealing** (Kirkpatrick et al., 1983; Geman and Geman, 1984; Laarhoven and Aarts, 1987) on a single chain.

simulated annealing

4.3.2.4 *Gibbs Sampling*

Gibbs sampler

Gibbs sampling (Geman and Geman, 1984) is a simple, efficient, and widely used sampling method based on the Metropolis-Hastings algorithm. Suppose we want to sample from a vector of random variables $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, and that the expressions for the full conditional distributions $P(x_j|\mathbf{x}_{-j}) = P(x_j|x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_m)$ are known. Each step of the Gibbs sampling involves replacing the value of one of the random variables by a value drawn from the distribution of that variable conditioned on the values of the remaining variables. More specifically, we replace the value of variable x_j by a value drawn from distribution $P(x_j|\mathbf{x}_{-j})$. This algorithm is repeated either by cycling through all random variables in a particular order or by choosing the order at random. This procedure is summarized in algorithm 4.3.

Algorithm 4.3 Gibbs sampling algorithm for N iterations.

```
Initialize x_j^(0) : j ∈ 1, 2, ..., m
i ← 0
while i < N do
    Sample x_1^(i+1) ~ P(x_1|x_{-1}^(i))
    Sample x_2^(i+1) ~ P(x_2|x_{-2}^(i))
    .
    .
    .
    Sample x_m^(i+1) ~ P(x_m|x_{-m}^(i))
    i ← i + 1
end while
```

To show that the Gibbs sampler generates samples from the distribution $P(\mathbf{x})$, we will show that the Gibbs sampler is a special case of the Metropolis-Hastings algorithm. Consider a Metropolis-Hastings sampling, where the proposal distribution is defined as

$$Q_j(\mathbf{x}^*|\mathbf{x}^{(i)}) = P(x_j^*|\mathbf{x}_{-j}^{(i)}). \tag{4.38}$$

Notice that $\mathbf{x}_{-j}^* = \mathbf{x}_{-j}^{(i)}$, because the components are unchanged by the sampling step, and also

$$P(\mathbf{x}^{(i)}) = P(x_j^{(i)}, \mathbf{x}_{-j}^{(i)}) = P(x_j^{(i)}|\mathbf{x}_{-j}^{(i)})P(\mathbf{x}_{-j}^{(i)}). \tag{4.39}$$

From these equations we can derive that the acceptance probability $\alpha(\mathbf{x}^{(i)} \to \mathbf{x}^*)$ is always one:

$$\begin{aligned}
\alpha(\mathbf{x}^{(i)} \to \mathbf{x}^*) &= \min\left(1, \frac{P(\mathbf{x}^*)Q_j(\mathbf{x}^*|\mathbf{x}^{(i)})}{P(\mathbf{x}^{(i)})Q_j(\mathbf{x}^{(i)}|\mathbf{x}^*)}\right) \\
&= \min\left(1, \frac{P(x_j^*|\mathbf{x}_{-j}^*)P(\mathbf{x}_{-j}^*)P(x_j^{(i)}|\mathbf{x}_{-j}^*)}{P(x_j^{(i)}|\mathbf{x}_{-j}^{(i)})P(\mathbf{x}_{-j}^{(i)})P(x_j^*|\mathbf{x}_{-j}^{(i)})}\right) \qquad (4.40) \\
&= \min(1, 1) \\
&= 1,
\end{aligned}$$

thus the Metropolis-Hastings steps are always accepted.

The practical applicability of the Gibbs sampler depends on the ease with which we are able to sample from the conditional distributions $P(x_i|\mathbf{x}_{-i})$. If the direct sampling from the conditional distributions is too hard or even impossible, we can draw samples using the Metropolis-Hastings algorithm embedded in the Gibbs sampler.

collapsed Gibbs sampler

An easy and widely used technique for speeding up the Gibbs sampler when our interest is only to sample from a subset of hidden variables is to use the **collapsed Gibbs sampler** (Liu, 1994). In the collapsed version of the Gibbs sampler we integrate out the unwanted variables when sampling from the other random variables.

conjugate prior

For example, imagine that we have three hidden variables x_1, x_2, x_3 and we only want to sample from x_1 and x_2. When generating a sample for x_1, we can draw the sample from the marginal distribution $P(x_1|x_2)$ with variable x_3 being integrated out. Likewise, the value of variable x_2 can be drawn from $P(x_2|x_1)$ with variable x_3 being integrated out, and not sample from x_3 at all. This is generally tractable when x_3 is the **conjugate prior** for x_1 and x_2. More details about conjugate priors will be discussed in chapter 5.

As we will see later, the collapsed Gibbs sampler is a good choice for the inference algorithm of the probabilistic semantic frames model proposed in this thesis.

5 SELECTED PROBABILITY DISTRIBUTIONS

In the previous chapter, we discussed probabilistic graphical models that will serve as a skeleton of the probabilistic frames models. Now, we turn to an exploration of some probability distributions that form building blocks of PGMs. They describe probability densities over random variables presented in the probabilistic graphical models. The list of distributions introduced in this chapter is limited to the basic distributions necessary for understanding the models described in further chapters.

The issue of choosing an appropriate probability distribution is a key problem in probability modeling and usually depends on the knowledge and intuition of the model designer. In this chapter, we will specifically discuss **parametric** distributions. They are governed by a small number of parameters that control the shape of a distribution under some restrictions given by the type of the distribution. These parameters are determined by an estimation from the data points. An alternative approach is known as a **non-parametric** model. The functional form of a non-parametric distribution depends moreover on the size of the data set. Non-parametric distributions will be discussed later concerning the context of the non-parametric model for probabilistic semantic frames.

parametric distribution

non-parametric distribution

An interesting concept that plays an important role in probabilistic models is known as **conjugate prior** (Raïffa and Schlaifer, 2000). In Bayesian probability theory, conjugate prior is a prior that is of the same family (functional form) as the corresponding posterior distribution. For example, the Gaussian distribution is the conjugate prior for its mean (it is so called **self-conjugate**), the conjugate prior for parameters of the multinomial distribution is the Dirichlet distribution, and the conjugate prior for the Poisson distribution is the Gamma distribution. The motivation for using a conjugate prior instead of a non-conjugate prior is a greatly simplified Bayesian analysis, as we shall see later.

conjugate prior

self-conjugate prior

In this chapter, we will explore binomial and multinomial distributions, representing the family of discrete distributions, together with their continuous conjugate priors, the beta and Dirichlet distributions. The examples from the family of continuous distributions shall be extended with the gamma distribution.

5.1 THE BINOMIAL AND THE MULTINOMIAL DISTRIBUTIONS

We begin to explore the family of discrete distributions by considering a binary random variable $x \in \{0, 1\}$. This variable, for example, can model the outcome of flipping a coin. Let us denote the probability of $x = 1$ by a constant p, where $0 \leqslant p \leqslant 1$. The probability distribution over x is known as the **Bernoulli** distribution and defined as

Bernoulli distribution

$$\mathrm{Bern}(x|p) = p^x(1-p)^{1-x}. \tag{5.1}$$

We can now draw n independent samples $\{x_1, x_2, \ldots, x_n\}$ from $\mathrm{Bern}(x \mid p)$ and ask how many times has happened that $x_i = 1$, which is denoted by $m \in \{0, 1, \ldots, n\}$. The distribution over the values of m is called the **binomial** distribution and is defined as

binomial distribution

$$\mathrm{Bin}(m|p, n) = \binom{n}{m} p^m (1-p)^{n-m}, \tag{5.2}$$

where

$$\binom{n}{m} = \frac{n!}{(n-m)!m!} \tag{5.3}$$

binomial coefficient

is the **binomial coefficient** expressing the number of ways of choosing m objects from a set of n objects. An example of the histogram of the binomial distribution for $n = 10$ and $p = 0.6$ is plotted in figure 5.1.

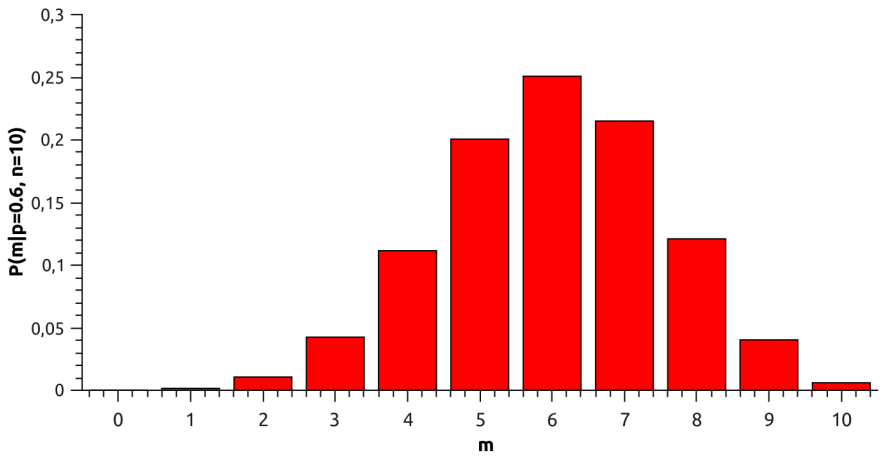


Figure 5.1: The binomial distribution with $p = 0.6$ and $n = 10$.

Notice that when $n = 1$, the binomial distribution is equal to the Bernoulli distribution.

Both discrete distributions that we have discussed heretofore are defined for binary variables. Now, let us generalize the concepts to k dimensions. We will replace the analogy of tossing a coin with tossing

a k-sided die. Let x be a k-dimensional random variable in which only one element is set to 1 and all the remaining elements are set to 0. For example, a valid value of a four-dimensional random variable $\mathbf{x}$ may be $\mathbf{x} = (0, 1, 0, 0)$. The probability distribution over all values of $\mathbf{x}$ is given by the **categorical** distribution defined as

categorical distribution

$$P(x|\mathbf{p}) = \prod_{i=1}^{k} p_i^{x_i}, \tag{5.4}$$

where $\mathbf{p} = (p_1, p_2, \ldots, p_k)$ is a real-valued vector satisfying $p_i \geqslant 0$ and $\sum_{i=1}^{k} p_i = 1$, which defines the probabilities of the outcomes of random variable $\mathbf{x}$. The categorical distribution can be seen as a generalization of the Bernoulli distribution to more than two outcomes.

Finally, we are able to generalize the binomial distribution to k dimensions. Let $\mathbf{m} = (m_1, m_2, \ldots, m_k)$ is a vector of the numbers of observations of all outcomes from sampling the random variable $\mathbf{x}$. The distribution over $\mathbf{m}$ takes the form

$$P(\mathbf{m}|\mathbf{p}) = \frac{(\sum_{i=1}^{k} m_i)!}{m_1! m_2! \ldots m_k!} \prod_{i=1}^{k} p_i^{m_i}, \tag{5.5}$$

which is known as the **multinomial** distribution.

multinomial distribution

5.2 THE BETA AND THE DIRICHLET DISTRIBUTIONS

All of the previously defined distributions have a parameter p or $\mathbf{p} = (p_1, p_2, \ldots, p_k)$ corresponding to probabilities of the outcomes of the random variable x. In order to develop a fully Bayesian analysis of the variable x, we need to introduce a prior over the parameter p. In other words, we need to define a probability distribution $P(p)$ for the Bernoulli and the binomial distributions and $P(p_1, p_2, \ldots, p_k)$ for the categorical and the multinomial distribution with dimension k.

Because of practical reasons, it is convenient to consider a form of the prior that has a simple interpretation as well as useful analytical properties. Those requirements are satisfied by the conjugate priors that have the same form as the likelihood function, thus the product of the prior and the likelihood will have the same form as well. This makes the Bayesian analysis much easier.

The conjugate prior for the Bernoulli distribution and the binomial distribution is called the **beta** distribution, and is given by

beta distribution

$$\mathrm{Beta}(p|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} p^{a-1}(1-p)^{b-1}, \tag{5.6}$$

where $a > 0$ and $b > 0$ are real-valued parameters controlling the shape of the distribution, and $\Gamma(x)$ is the **gamma** function defined as

gamma function

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du. \tag{5.7}$$

Using integration by parts, it can be proven that $\Gamma(x+1) = x\Gamma(x)$. This implies that $\Gamma(1) = 1$ and $\Gamma(x+1) = x!$ when x is a positive integer (Bishop, 2007). That is why the gamma function can be seen as a real-valued extension of the factorial function $x!$ with the argument x shifted down by 1.

The gamma function ensures that the beta distribution is normalized. How the parameters a and b control the shape of the distribution is shown in figure 5.2.

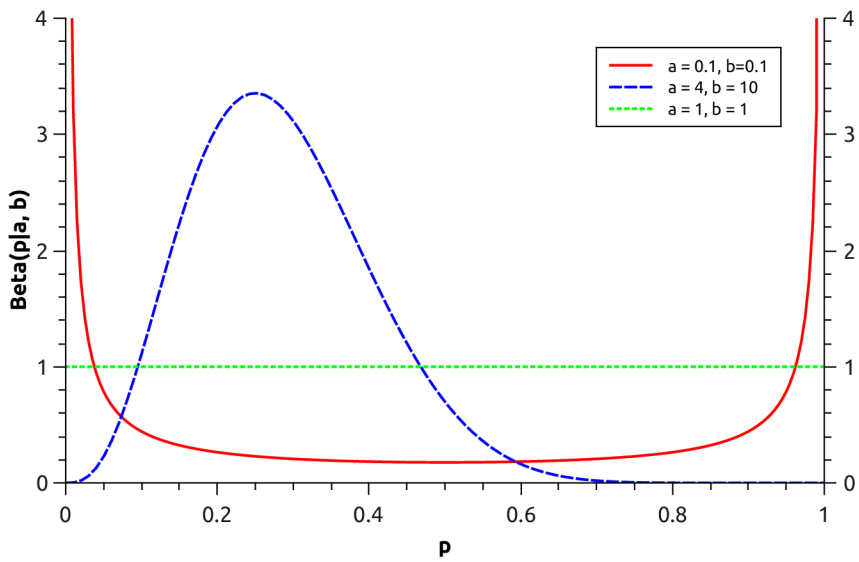


Figure 5.2: The beta distribution for various values of the parameters.

Notice that while the distribution has one peak when the parameters are greater than one, when the parameters are equal to 0.1, the density has the highest values around both extreme values $p = 0$ and $p = 1$. The analogy with tossing a coin is that there is a high prior probability that the coin is unfair, but we do not know which side of the coin is preferred. This is a very important property with the beta distribution, which has a big impact on probabilistic topic models as we will see later on.

The posterior density over the probability p can now be obtained by multiplying the beta prior and the binomial likelihood and normalizing. When we skip the factors that do not depend on p, the posterior density is proportional to

$$\begin{aligned} P(p|m, n, a, b) &\propto p^{a-1}(1-p)^{b-1} p^m (1-p)^{n-m} \\ &= p^{m+a-1}(1-p)^{n-m+b-1}, \end{aligned} \tag{5.8}$$

which is simply another beta distribution, so the normalization coefficient can be obtained similarly to 5.6:

$$P(p|m,n,a,b) = \frac{\Gamma(m+a+n-m+b)}{\Gamma(m+a)\Gamma(n-m+b)} p^{m+a-1}(1-p)^{n-m+b-1}. \tag{5.9}$$

We can see from 5.9 that the parameters a and b can intuitively be interpreted as pseudo-counts of $x = 1$ and $x = 0$, respectively.

Now, we can introduce an extension of the beta distribution, which models a prior probability over the parameters of the multinomial distribution $\mathbf{p} = (p_1, p_2, \ldots p_k)$. The conjugate prior for the multinomial distribution is known as the **Dirichlet** distribution, and is given by

Dirichlet distribution

$$\mathrm{Dir}(\mathbf{p}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_1 + \alpha_2 + \cdots + \alpha_k)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\ldots\Gamma(\alpha_k)} \prod_{i=1}^{k} p_i^{\alpha_i - 1}, \tag{5.10}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ are parameters of the distribution. Since $0 \leqslant p_i \leqslant 1$ and $\sum_{i=1}^{k} p_i = 1$ must hold, the distribution over $\mathbf{p}$ lies on a **simplex** of a dimensionality $k-1$. Four examples of a three-dimensional Dirichlet distribution are depicted in figure 5.3[1].

simplex

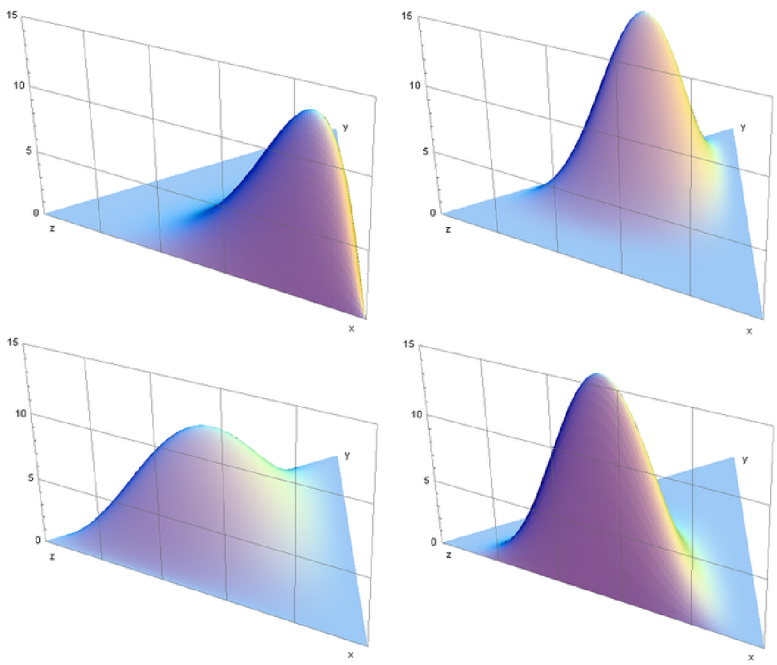


Figure 5.3: Examples of three-dimensional Dirichlet distributions.

1 The figure is adopted from http://en.wikipedia.org/wiki/Dirichlet_distribution.

By multiplying the Dirichlet distribution (prior) with the multinomial distribution (likelihood), we get the posterior distribution over parameters $\mathbf{p} = (p_1, p_2, \ldots, p_k)$:

$$P(\mathbf{p}|\mathbf{m}, \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k + m_1 + \cdots + m_k)}{\Gamma(\alpha_1 + m_1) \ldots \Gamma(\alpha_k + m_k)} \prod_{i=1}^{k} p_i^{\alpha_i + m_i - 1}, \quad (5.11)$$

which again takes the form of a Dirichlet distribution.

5.3 THE GAMMA DISTRIBUTION

gamma distribution

The **gamma** distribution is a continuous probability distribution with two real-number parameters $a > 0$ and $b > 0$, which is defined as:

$$\text{Gamma}(x|a, b) = \frac{1}{\Gamma(a)} b^a x^{a-1} e^{-bx}, \quad (5.12)$$

shape parameter
rate parameter

where a is usually called the **shape parameter** and b is called the **rate parameter**. Its density function for various choices of parameters a and b is depicted in figure 5.4. The gamma distribution simply stems

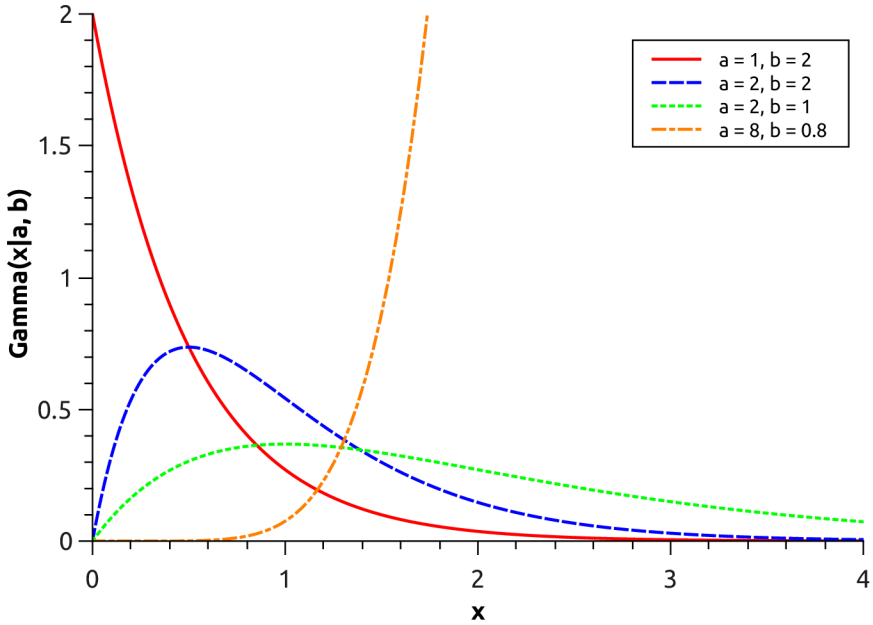


Figure 5.4: Probability density functions for various parameters of the gamma distribution.

from the gamma function, defined in 5.7. It is derived by normalizing the gamma function to have its integral equal to one:

$$\frac{\Gamma(a)}{\Gamma(a)} = \int_0^\infty \frac{x^{a-1} e^{-x}}{\Gamma(a)} dx, \quad (5.13)$$

and hence

$$\text{Gamma}(x|a, 1) = \frac{1}{\Gamma(a)} x^{a-1} e^{-x}, \tag{5.14}$$

which can be viewed as a special case of scaled gamma distribution defined in 5.12.

The gamma distribution is also closely related to the **exponential** distribution, defined as

exponential distribution

$$\text{Exponential}(x|\lambda) = \lambda e^{-\lambda x}, \tag{5.15}$$

which simply is a special case of the gamma distribution with $a = 1$ and $b = \lambda$.

Thanks to its form, the gamma distribution is often used as a prior distribution for many likelihood functions, namely the Poisson, exponential or the normal distribution with known mean.

6 TOPIC MODELS

The core idea behind LDA-Frames is inspired by topic models, originally proposed to understand the content of a text document. The topics can be seen as abstract themes that occur in documents, and can be used to summarize the said documents. For example *soccer, sport, Europe* and *news* can be topics describing a web document reporting the latest results of soccer in Europe.

The first reference to topic models, which comes from Information Retrieval (IR), is dated back to the nineteen-eighties, and is known as **Latent Semantic Indexing** (Dumais et al., 1988; Furnas et al., 1988). Most approaches for retrieving textual documents before Latent Semantic Indexing (LSI) were based on lexical matching between words in user's queries and a collection of documents. Because there are many ways of expressing a user's query, and a lot of words are ambiguous, the literal matching may not find relevant documents or even may find irrelevant documents. A better way to retrieve documents seems to be matching conceptual topics that represent the meaning of the query and the documents. That is what LSI does. More specifically, LSI tries to overcome those problems by modeling a high-dimensional space of words using a low-dimensional space of topics.

Latent Semantic Indexing

Both documents and queries in LSI are represented with a subset of topics taken from a shared collection of general topics, where each topic has its own weight of membership for a given document. Similarly, the topics are represented as subsets of the vocabulary with the weights assigned to selected words. In the original LSI, the weights may have negative values, which indicate negative relevance. For example

(6.1) *soccer* = 1.8*'soccer' + 0.4*'ball' + 0.2*'FIFA' - 0.4*'tennis'

may be a representation of the *soccer* topic, and

(6.2) **doc** = *2.3*soccer + 1.8*sport + 0.9*Europe + 0.8*news*

a representation of the document **doc** describing the web document from the previous example.

The topic and document representations are derived from a word-document co-occurrence matrix using a dimensionality reduction technique. It is important that the process of topic identification is

completely unsupervised, and the topic labels (e.g. 'Soccer' or 'News') are unknown. In most IR applications, however, there is usually no need for labels, thus they can be replaced with integers.

Topics generated with LSI techniques can be considered as semantic classes identified from the vocabulary. This fact has been an inspiration for using topics as the representation of semantic roles in our model for probabilistic semantic frames. Before we describe the LDA-Frames model, we will introduce three approaches for topic modeling. The first one is called **Latent Semantic Analysis**, which is often used as a synonym for LSI. It is based on linear algebra techniques. The other two models are probabilistic, and are much closer to our model for semantic roles. Their names are **Probabilistic Latent Semantic Analysis** and **Latent Dirichlet Allocation**.

Latent Semantic Analysis

Probabilistic Latent Semantic Analysis

Latent Dirichlet Allocation

6.1 LATENT SEMANTIC ANALYSIS

Latent Semantic Analysis (Dumais et al., 1988; Furnas et al., 1988), also known as Latent Semantic Indexing, is a technique that projects documents into a vector space with latent semantic dimensions, which is typically much lower than the original space. Thus, we can look at Latent Semantic Analysis (LSA) as a dimensionality reduction technique. If two words often co-occur in the same documents, they are likely semantically similar. That is the co-occurrence on the first level. Moreover, if two words often co-occur with a third word in the same documents, they can also be considered to be semantically related. That is the co-occurrence on the second level. We can define the co-occurrence on the third level, fourth level, and so on. The notion of semantic similarity is used to group words into topics.

LSA is an application of a method from linear algebra, called **Singular Value Decomposition**, to a word-by-document matrix. The projection into a latent semantic space is used such that the 2-norm between the original co-occurrence matrix and the low-dimensional latent space matrix is minimized. Let us start with a sample collection of three one-sentenced documents:

Singular Value Decomposition

(6.3) $\mathbf{Doc}_1$: *Machine learning helps people to understand data.*

(6.4) $\mathbf{Doc}_2$: *Data can be understood using machine learning.*

(6.5) $\mathbf{Doc}_3$: *People can use machine learning for data understanding.*

We can extract the vocabulary as an alphabetically ordered list of all lower-cased words used in the documents

$\mathbf{V}$ = ('be', 'can', 'data', 'for', 'helps', 'learning', 'machine', 'people', 'to', 'understand', 'understanding', 'understood', 'use', 'using'),

and transform the documents into a word-by-document matrix, also known as the **bag-of-words** representation, which stores the word-count frequencies for every word from the vocabulary in the documents. The bag-of-words representation for our three documents is illustrated in table 6.1.

bag-of-words representation

	$\mathbf{Doc}_1$	$\mathbf{Doc}_2$	$\mathbf{Doc}_3$
be	0	1	0
can	0	1	1
data	1	1	1
for	0	0	1
helps	1	0	0
learning	1	1	1
machine	1	1	1
people	1	0	1
to	1	0	0
understand	1	0	0
understanding	0	0	1
understood	0	1	0
use	0	0	1
using	0	1	0

Table 6.1: The bag-of-words representation of text documents.

Let us denote the word-by-document matrix $\mathbf{M}$. We decompose this matrix using the Singular Value Decomposition (SVD) to obtain a product of three matrices

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}}, \tag{6.1}$$

where $\mathbf{U}$ is a column matrix of the left singular vectors for words, $\mathbf{V}^{\mathsf{T}}$ is the transpose of a column matrix of the right singular vectors for documents, and $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values. The difference between the original matrix $\mathbf{M}$ and the decomposition $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}}$ is that the decomposition comprises orthogonal eigen vectors with their eigen values, thus it removes redundancy. When we remove the singular triplet (a pair of vectors from $\mathbf{U}$ and $\mathbf{V}^{\mathsf{T}}$ with associated singular value from $\mathbf{\Sigma}$) with the lowest singular value, it defines the best lower-rank approximation $\mathbf{M}'$ of the original matrix $\mathbf{M}$ such that

$$\Delta = \|\mathbf{M} - \mathbf{M}'\|_2 \tag{6.2}$$

is minimal (Eckart and Young, 1936).

The LSA method is based on discarding $|\mathbf{V}| - k$ singular triplets with the lowest singular values ($k \cong 300$ for big corpora, typically) in the SVD decomposition of the word-by-document matrix, which results in a representation of the original documents in k-dimensional latent topic space.

Specifically, the decomposition of matrix $\mathbf{M}$ is given by matrices $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$ as follows.

$$\mathbf{U} = \begin{pmatrix} -1.4109e-01 & 3.5355e-01 & 2.5185e-01 \\ -3.0845e-01 & 3.5355e-01 & -1.7280e-01 \\ -4.4954e-01 & -1.0126e-16 & 7.9044e-02 \\ -1.6735e-01 & 4.2962e-16 & -4.2465e-01 \\ -1.4109e-01 & -3.5355e-01 & 2.5185e-01 \\ -4.4954e-01 & -1.0126e-16 & 7.9044e-02 \\ -4.4954e-01 & -1.0126e-16 & 7.9044e-02 \\ -3.0845e-01 & -3.5355e-01 & -1.7280e-01 \\ -1.4109e-01 & -3.5355e-01 & 2.5185e-01 \\ -1.4109e-01 & -3.5355e-01 & 2.5185e-01 \\ -1.6735e-01 & 4.2962e-16 & -4.2465e-01 \\ -1.4109e-01 & 3.5355e-01 & 2.5185e-01 \\ -1.6735e-01 & 4.2962e-16 & -4.2465e-01 \\ -1.4109e-01 & 3.5355e-01 & 2.5185e-01 \end{pmatrix} \tag{6.3}$$

$$\mathbf{\Sigma} = \begin{pmatrix} 3.8399 & 0 & 0 \\ 0 & 2.0000 & 0 \\ 0 & 0 & 1.8043 \end{pmatrix} \tag{6.4}$$

$$\mathbf{V}^T = \begin{pmatrix} -5.4177e-01 & -7.0711e-01 & 4.5440e-01 \\ -5.4177e-01 & 7.0711e-01 & 4.5440e-01 \\ -6.4262e-01 & 8.6820e-16 & -7.6618e-01 \end{pmatrix} \tag{6.5}$$

A corpus of three documents cannot produce more than three singular triplets, so both the documents and the words are represented as three-dimensional vectors. We can reduce the space, for example into two dimensions, by removing the singular triplet with the singular value 1.804. After doing that, we get the representation of the documents in a two-dimensional space of latent topics given by table 6.2.

To summarize, LSA creates a latent semantic space in which documents and words can be compared. The dimensionality of the semantic space is typically much lower than the number of words and

	Doc_1	Doc_2	Doc_3
$topic_1$	-5.4177e-01	-5.4177e-01	-6.4262e-01
$topic_2$	-7.0711e-01	7.0711e-01	8.6820e-16

Table 6.2: Reduced 2-dimensional representation of the documents.

documents, so it can be seen as a dimensionality reduction method. The method is unsupervised and the latent topics have no explicit labels derived using the LSA procedure.

6.2 PROBABILISTIC LATENT SEMANTIC ANALYSIS

Although LSA has been applied with remarkable success in different domains, it has a number of limitations mainly stemming from its non-adequate statistical foundation. In order to overcome the deficits, Hofmann (1999) developed a fully statistical approach called Probabilistic Latent Semantic Analysis (PLSA), also known as the **Aspect Model**. The PLSA approach models each word in a document as a sample from a mixture model. The mixture components are multinomial random variables representing latent topics, such that each word is generated from a single topic, and a single document can be associated with multiple topics. A document can thus be viewed as a probability distribution on a fixed set of topics.

Aspect Model

The idea of probabilistic topic models is illustrated in figure 6.1, which is adopted from Steyvers and Griffiths (2007).

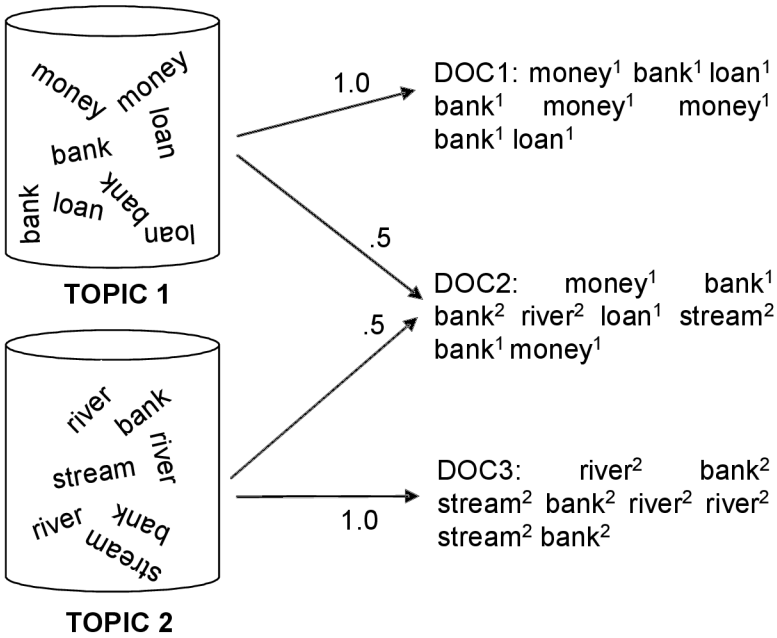


Figure 6.1: Generative process for probabilistic topic models.

The generative probabilistic model in the figure describes a way of generating documents from known topics and probability distributions over them. Topics 1 and 2 are thematically related to money and rivers, respectively, and are illustrated as a bag of words, where multiple occurrences of words simulate different probabilities. Particular documents can be produced by picking words depending on the probability given to the topic. For example, documents 1 and 3 were generated from topics 1 and 2, respectively, whereas document 2 was generated from both topics 1 and 2 with an equal probability. The superscript numbers associated with the generated words indicate topics that were used to sample those words. This allows probabilistic topic models to treat with polysemy where the same words have multiple meanings.

Formally speaking, PLSA model associates a latent topic variable $z \in \{z_1, z_2, \ldots z_K\}$ with each occurrence of a word $w \in \{w_1, w_2, \ldots, w_M\}$ in a document $d \in \{d_1, d_2, \ldots, d_N\}$, which can be described in terms of the generative process defined in algorithm 6.1.

Algorithm 6.1 Generative process for PLSA.

```
for i ∈ {1, 2, ..., N} do
    for j ∈ {1, 2, ..., M} do
        Choose a latent topic z_ij with probability P(z_ij|d_i)
        Choose a word w_ij with probability P(w_ij|z_ij)
    end for
end for
```

Notice that the number of words M in document d_i is the same for each $i \in \{1, 2, \ldots, N\}$. This is only a simplification used to ease the notation and can be easily overcome. It is worth mentioning that samples of topics and words in a document are independent of each other. The generative process can also be represented using a graphical probabilistic model, as is shown in figure 6.2.

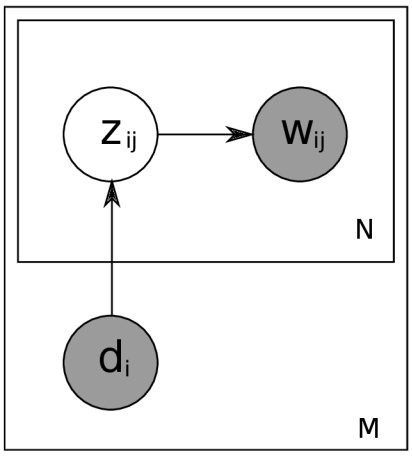


Figure 6.2: Graphical model for PLSA.

The crucial issue with modeling documents using PLSA is the inference of latent topic variables $\boldsymbol{z}$. We start with expressing the joint distribution $P(\mathbf{d}, \boldsymbol{w})$, where the observation pairs (d, w) are supposed to be generated independently, and words w conditioned on a latent variable z are conditionally independent of the specific document d. This results in the following expressions:

$$P(d, w) = P(d)P(w|d), \tag{6.6}$$

where

$$P(w|d) = \sum_{k=1}^{K} P(w|z_k)P(z_k|d). \tag{6.7}$$

After the application of the Bayes' rule on $P(z_k|d)$ we get

$$P(d, w) = \sum_{k=1}^{K} P(z_k)P(w|z_k)P(d|z_k). \tag{6.8}$$

The log likelihood of the data is then given by

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{j=1}^{M} \log P(d_i, w_{ij}). \tag{6.9}$$

The standard procedure to estimate the maximum likelihood is to use the Gibbs sampling or the variational inference. In this case, the variational inference can be replaced by its simpler variant called the Expectation-Maximization (EM) algorithm, which alternates between performing an expectation (E) step in order to create a function for the expected values of the latent variables $\boldsymbol{z}$ using the current estimates of the parameters, and a maximization (M) step, which computes the parameters by maximizing the expected log likelihood from the E step.

Following the original work of Hofmann (1999), the expectation over the variable z in the E step is

$$P(z|d, w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{k=1}^{K} P(z_k)P(d|z_k)P(w|z_k)}, \tag{6.10}$$

and the M step equations are given by

$$P(w|z) = \frac{\sum_{i=1}^{N} n(d_i, w)P(z|d_i, w)}{\sum_{i=1}^{N} \sum_{j=1}^{M} P(z|d_i, w_{ij}))}, \tag{6.11}$$

$$P(d|z) = \frac{\sum_{j=1}^{M} P(z|d, w_{dj})}{\sum_{i=1}^{N} \sum_{j=1}^{M} P(z|d, w_{ij})}, \tag{6.12}$$

$$P(z) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} P(z|d_i, w_{ij})}{\sum_{i=1}^{N} \sum_{j=1}^{M} n(d_i, w_{i,j})}, \tag{6.13}$$

where $n(d, w)$ denotes the number of times word w occurred in document d. Alternating between the E and M steps creates a procedure that approaches a local maximum of the log likelihood function defined in equation 6.9.

The PLSA model performs considerably better than LSA in many applications (Hofmann, 1999), but suffers from two problems. First, the topic mixtures $P(z|d)$ are learned only for the training documents that have been seen during the learning phase, therefore there is no way to estimate those distributions for previously unseen documents. This property of PLSA limits its areas of applications. Secondly, there are a lot of parameters needed to be learned. Particularly, the number of parameters is given by $kV + kN$, where k is the number of topics, V is the size of vocabulary, and N is the number of documents. It especially means that the number of parameters grows linearly with the number of documents, which leads to overfitting for large datasets (Popescul et al., 2001). To overcome all these drawbacks Blei et al. (2003) proposed a model called **Latent Dirichlet Allocation**.

Latent Dirichlet Allocation

6.3 LATENT DIRICHLET ALLOCATION

We have seen that the PLSA model does not make any prior assumptions about the mixture weights of topics in documents, nor about the distribution of words in topics. This makes it difficult to generalize the model to new documents. Being inspired by PLSA, Blei et al. (2003) extended the model by introducing a Dirichlet prior on both topic and word distributions, and named the resulting generative model Latent Dirichlet Allocation (LDA).

Let us remind the definition of the Dirichlet distribution:

$$\mathrm{Dir}(\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_1 + \alpha_2 + \cdots + \alpha_K)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\ldots\Gamma(\alpha_K)} \prod_{i=1}^{K} p_i^{\alpha_i - 1} \tag{6.14}$$

It has been shown that the PLSA model is equivalent to the LDA model with uniform Dirichlet priors ($\alpha_1 = \alpha_2 = \cdots = \alpha_k = 1$) (Girolami and Kabán, 2003), however, setting the $\boldsymbol{\alpha}$ weights reasonably results in a more natural model of text documents. Specifically, for $\alpha_1 = \alpha_2 = \cdots = \alpha_k < 1$, the modes of the Dirichlet distribution are located at the corners of the simplex. In this regime, there is a bias towards sparsity, and a pressure to pick topic distributions favoring only a few topics. The same regime is often used for the word distribution prior controlled with hyperparameters $\boldsymbol{\beta}$. The particular values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are usually set to $\alpha_1 = \alpha_2 = \ldots \alpha_k = \beta_1 = \beta_2 = \cdots = \beta_V = 0.1$ in practice. A formal description of the generative process for LDA is written down in algorithm 6.2.

The graphical model represented in figure 6.3 reveals a multilevel structure of the LDA assumptions. While the mixture proportions of topics are drawn uniquely for each document with a prior shared for

Algorithm 6.2 Generative process for LDA.

```
for i ∈ {1, 2, ..., N} do
    Choose θ_i from Dir(α)
    for j ∈ {1, 2, ..., M} do
        Choose a latent topic z_ij from Multinomial(θ_i)
        Choose a word w_ij from Multinomial(φ_{z_ij})
                with a Dirichlet prior Dir(β)
    end for
end for
```

the whole collection of documents, the mixture components (topic themselves) are shared across the collection with their priors shared as well.

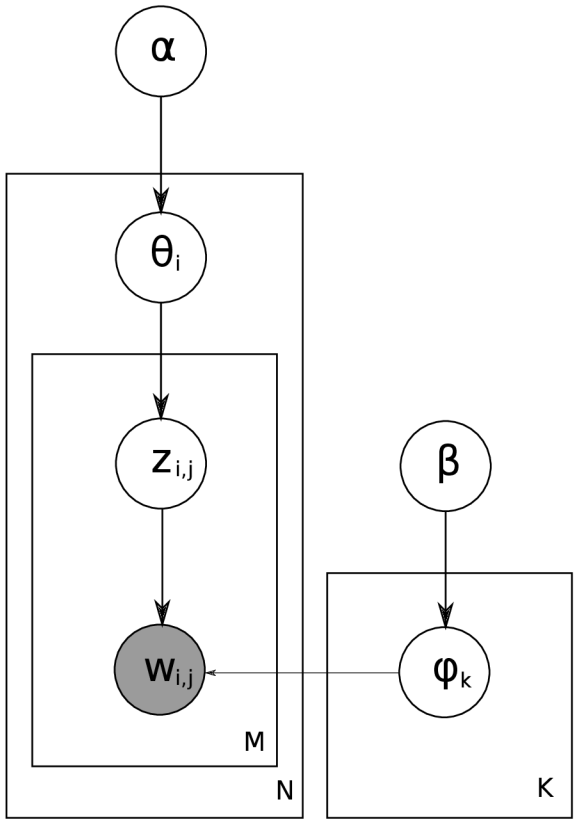


Figure 6.3: Graphical model of latent Dirichlet allocation.

Apart from the hyperparameters α and β, the number of topics K also requires to be defined in advance within this basic LDA model. The mixing coefficients of topics for each document θ, and the word-topic distributions φ, are hidden, and are learned from the observed data using unsupervised learning techniques. For learning the hidden parameters of LDA, Blei et al. (2003) introduced a variational inference algorithm. Subsequently, Griffiths and Steyvers (2004) developed an

algorithm based on collapsed Gibbs sampling, which generates a sequence of samples from the joint probability distribution of the LDA model. Whereas the variational inference is faster (it only needs to see the observed data once), the collapsed Gibbs sampling gives more accurate results.

Given the observed data $\mathbf{w} = \{w_{ij}\}$, the task of the Bayesian inference is to compute the posterior distribution over the latent topics $\mathbf{z} = \{z_{ij}\}$, where the posterior distribution of the whole model is given by

$$P(\mathbf{w}, \mathbf{z}, \theta, \boldsymbol{\varphi} | \alpha, \beta) = \prod_{k=1}^{K} \mathrm{Dir}(\boldsymbol{\varphi}_k | \beta) \prod_{i=1}^{N} \mathrm{Dir}(\theta_i | \alpha) \prod_{j=1}^{M} \mathrm{Mult}(z_{ij} | \theta_i) \mathrm{Mult}(w_{i,j} | \boldsymbol{\varphi}_{z_{ij}}). \tag{6.15}$$

Generative probabilistic models for topic modeling give great potential to help understand natural languages. These models make explicit assumptions about the process of generating a document, and enable the identification of latent structures that underlie the words in documents. In this thesis, their advantages are exploited to make the core of the probabilistic semantic frames named LDA-Frames.

7 LDA-FRAMES

In this chapter, we will propose an unsupervised model for probabilistic semantic frames named LDA-Frames. It is based on my previous work published in Materna (2012a). Instead of assuming the availability of annotated data with known frames and roles, we rely on automatically generated syntactic dependency information. More specifically, for every occurrence of a predicate (lexical unit) from the set of predefined predicates in the training corpus, we automatically identify realizations for a list of predefined grammatical relations. For example, suppose that the set of the predefined predicates is given by

$$\text{PRED} = \{\texttt{eat}, \texttt{drink}, \texttt{teach}\}, \tag{7.1}$$

the list of grammatical relations is given by

$$\text{REL} = [\texttt{subject}, \texttt{object}], \tag{7.2}$$

and the corpus only consists of one sentence

(7.1) *Thomas eat a whole pie and drink a pint of milk for breakfast.*

If subject represents the subject of the predicate and object represents the direct object of the predicate, the algorithm is supposed to generate training data depicted in table 7.1.

predicate	**subject**	**object**
eat	Thomas	pie
drink	Thomas	pint

Table 7.1: Example of training data.

Based on the training data, we treat the semantic frame induction problem as a clustering task. In the first stage, the goal is to assign each training example (a row in table 7.1) to a cluster, so that every cluster of training data consists of instances assigned to the same semantic frame. In the second stage, we assign each cluster a frame specification represented as a tuple of semantic roles, so that each tuple is unique across all clusters. Semantic roles as well as semantic frame clusters, can be represented with any unique identifiers, typically with positive integers.

predicate	subject	object	semantic frame	semantic roles
eat	Thomas	pie	1	(1, 3)
	John	food		
	Mike	pizza		
	man	cake		
	dog	meat	2	(2, 3)
	mouse	cheese		
	cat	mouse		
drink	Thomas	pint	3	(1, 4)
	Jane	coffee		
	Mike	tee		
teach	teacher	student	4	(1, 1)
	professor	Mike		
	Peter	dog	5	(1, 2)

Table 7.2: Extended example of a training data set.

In order to exemplify the idea, let us extend the data set from table 7.1 with other training examples. The extended data set with some additional columns is shown in table 7.2.

You can see that the training examples are clustered according to their semantic frame identifiers shown in the third column. Each frame cluster is then enhanced with a tuple of semantic roles, printed out in the fourth column. Although the semantic role identifiers may not necessarily be associated with explicit labels, one can easily recognize a straightforward assignment of semantic role labels for the example data, which is shown in table 7.3.

semantic role ID	semantic role label
1	Person
2	Animal
3	Food
4	Drink

Table 7.3: Semantic role labels

The semantic frame and semantic role assignments naturally define probability distributions over semantic frames (SF) for a predicate, and probability distributions over semantic role realizations (SRR) for a semantic role (SR). For instance,

$$P(\mathrm{SF} = 2 \,|\, \mathrm{predicate} = \mathrm{eat}) = \frac{3}{7}, \tag{7.3}$$

$$P(\mathrm{SRR} = \mathrm{Mike} \,|\, \mathrm{SR} = 1) = \frac{3}{12} = \frac{1}{4}. \tag{7.4}$$

These distributions can further be exploited in an automatic annotation of previously unseen texts with the inferred semantic frames and roles.

The two phases of semantic frame and semantic role induction need not be performed separately, nor in any specific order. Their implementation, proposed in this thesis, will be clarified in the subsequent sections.

7.1 MODEL DESCRIPTION

The method for automatic identification of semantic frames is based on a probabilistic generative process. Training data for the algorithm consists of tuples of grammatical relation realizations acquired using a dependency parser from a training corpus for every predicate (lexical unit). Each grammatical relation realization is treated as generated from a given semantic frame according to the word distribution of the corresponding semantic role.

Formally, let

$$\mathbf{g} = (g_1, g_2, \ldots, g_S) \tag{7.5}$$

be a list of selected grammatical relations,

$$\mathbf{lu} = (lu_1, lu_2, \ldots, lu_U) \tag{7.6}$$

list of lexical units, and

$$\begin{aligned}
\boldsymbol{w} = & \\
& \{ \\
& \{(w_{1,1,1}, w_{1,1,2}, \ldots, w_{1,1,S}), \ldots, (w_{1,T,1}, w_{1,T,2}, \ldots, w_{1,T,S})\}, \\
& \{(w_{2,1,1}, w_{2,1,2}, \ldots, w_{2,1,S}), \ldots, (w_{2,T,1}, w_{2,T,2}, \ldots, w_{2,T,S})\}, \\
& . \\
& . \\
& . \\
& \{(w_{U,1,1}, w_{U,1,2}, \ldots, w_{U,1,S}), \ldots, (w_{U,T,1}, w_{U,T,2}, \ldots, w_{U,T,S})\} \\
& \}
\end{aligned} \tag{7.7}$$

grammatical relation realizations of training data from the corpus. For simplicity, we assume that each lexical unit has T occurrences in the corpus. This simplification can be easily avoided by indexing $T_1, T_2, \ldots T_U$ for lexical units $u \in \{1, 2, \ldots, U\}$.

Supposing the number of frames is given by parameter F, the number of semantic roles by R, the number of slots (grammatical relations) by S and the size of vocabulary is V. The generative process is defined according to algorithm 7.1.

Algorithm 7.1 Generative process for LDA-Frames.

1: **for** $f \in \{1, 2, \ldots, F\}$ **do**
2: **do**
3: **for** $s \in \{1, 2, \ldots, S\}$ **do**
4: Choose $r_{f,s}$ from Uniform$(1, R)$
5: **end for**
6: **while** the frame is not unique
7: **end for**
8: **for** $r \in \{1, 2, \ldots, R\}$ **do**
9: Choose θ_r from Dir(β)
10: **end for**
11: **for** $u \in \{1, 2, \ldots, U\}$ **do**
12: Choose φ_u from Dir(α)
13: **for** $t \in \{1, 2, \ldots, T\}$ **do**
14: Choose frame $f_{u,t}$ from Multinomial(φ_u),
15: where $f_{u,t} \in \{1, 2, \ldots, F\}$.
16: **for** $s \in \{1, 2, \ldots, S\}$ **do**
17: Choose $w_{u,t,s}$ from Multinomial$(\theta_{r_{f_{u,t},s}})$
18: **end for**
19: **end for**
20: **end for**

Lines 1 up to 7 of the algorithm are responsible for generating a set of semantic frames, where each frame is in the form of a tuple of S semantic roles represented with their integer identifiers. The roles are chosen from the uniform distribution over integers between 1 and R. If a generated frame is identical with another frame from a previous iteration, sampling is repeated. This results in a set of F distinct frames. In order to make it possible to generate a set of distinct frames, the number of frames, roles and slots must fulfill the following formula:

$$R \geqslant \left\lceil \exp\left(\frac{\log(F)}{S}\right)\right\rceil . \tag{7.8}$$

Since it is not guaranteed that the do-while cycle terminates after a finite number of iterations, a better solution is to allow to generate only unique frames, which is a simple extension.

The semantic role realization distributions for all semantic roles are generated at lines 8 up to 10. The probabilities are chosen from the Dirichlet distribution with parameter β. Although the Dirichlet

distribution allows to define distinct weights for its dimensions, it is often convenient to use the symmetric Dirichlet distribution, where

$$\beta_1 = \beta_2 = \cdots = \beta_R. \tag{7.9}$$

Finally, the corpus realizations are generated at lines 11 up to 20. The distribution φ_u over semantic frames for a lexical unit u is generated from the Dirichlet distribution with parameter α. Similarly to θ, it is also useful to use the symmetric Dirichlet distribution. Semantic role realizations (particular words from training data) are generated from the distribution θ, where the semantic role index is chosen based on the sampled frame $f_{u,t}$, generated from the multinomial distribution with parameter φ_u.

The graphical model for LDA-Frames is shown in figure 7.1. It is parametrized with hyperparameters of prior distributions α and β, usually set by hand to a value between 0.01 – 0.1.

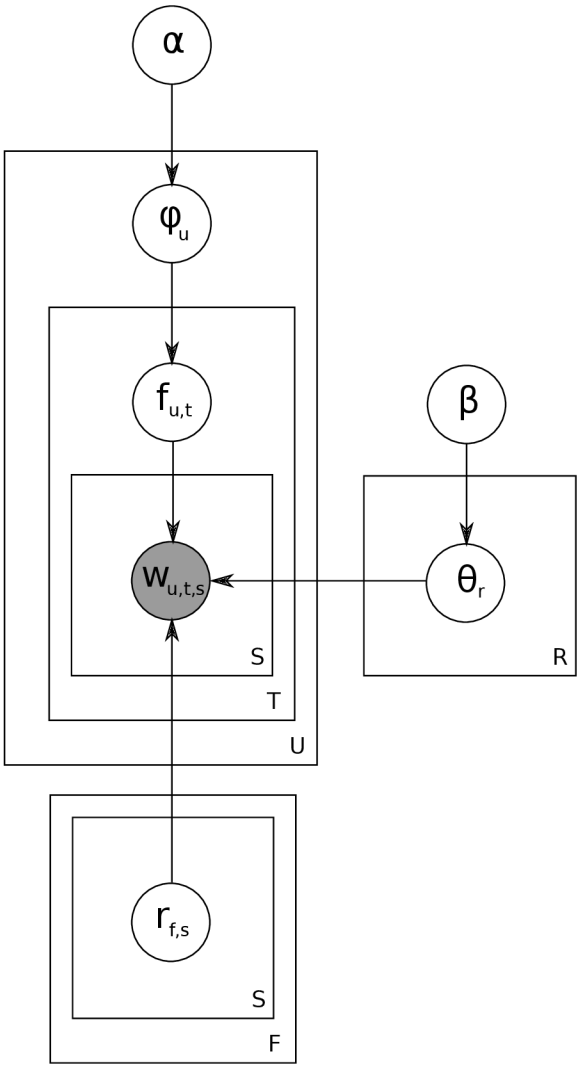


Figure 7.1: Graphical model for LDA-frames.

7.2 THE GIBBS SAMPLER FOR LDA-FRAMES

With the knowledge of the observed grammatical relation realizations $\boldsymbol{w}$, the task of the inference is to generate a set of semantic frames and compute the posterior distributions for frames and roles, $\boldsymbol{\varphi}$ and θ, respectively. It can be solved using the variational inference method or using a sampling algorithm.

Based on the arguments from chapter 4, we will focus on explaining and implementing the collapsed Gibbs sampler, which seems to be the most suitable inference method for LDA-frames.

For the implementation of the collapsed Gibbs sampler, we need to express conditional probabilities for hidden variables

$$P(f_{u,t}|\mathbf{f}_{-(\mathbf{u},\mathbf{t})}, \mathbf{r}, \boldsymbol{w}, \alpha, \beta) \tag{7.10}$$

and

$$P(r_{f,s}|\mathbf{r}_{-(\mathbf{f},\mathbf{s})}, \mathbf{f}, \boldsymbol{w}, \alpha, \beta), \tag{7.11}$$

where $\mathbf{f}_{-(\mathbf{u},\mathbf{t})}$ denotes all random variables $f_{i,j}$ excluding such that $i = u, j = t$, and $\mathbf{r}_{-(\mathbf{f},\mathbf{s})}$ denotes all random variables $r_{i,j}$ except of $r_{f,s}$.

By the definition of conditional probability

$$P(f_{u,t}|\mathbf{f}_{-(\mathbf{u},\mathbf{t})}, \mathbf{r}, \boldsymbol{w}, \alpha, \beta) = \frac{P(f_{u,t}, \mathbf{f}_{-(\mathbf{u},\mathbf{t})}, \mathbf{r}, \boldsymbol{w}|\alpha, \beta)}{P(\mathbf{f}_{-(\mathbf{u},\mathbf{t})}, \mathbf{r}, \boldsymbol{w}|\alpha, \beta)} \tag{7.12}$$

and

$$P(r_{f,s}|\mathbf{r}_{-(\mathbf{f},\mathbf{s})}, \mathbf{f}, \boldsymbol{w}, \alpha, \beta) = \frac{P(r_{f,s}, \mathbf{r}_{-(\mathbf{f},\mathbf{s})}, \mathbf{f}, \boldsymbol{w}|\alpha, \beta)}{P(\mathbf{r}_{-(\mathbf{f},\mathbf{s})}, \mathbf{f}, \boldsymbol{w}|\alpha, \beta)} \tag{7.13}$$

We can remove the denominator from 7.12, which does not depend on $f_{u,t}$

$$\begin{aligned} &\frac{P(f_{u,t}, \mathbf{f}_{-(\mathbf{u},\mathbf{t})}, \mathbf{r}, \boldsymbol{w}|\alpha, \beta)}{P(\mathbf{f}_{-(\mathbf{u},\mathbf{t})}, \mathbf{r}, \boldsymbol{w}|\alpha, \beta)} \propto P(f_{u,t}, \mathbf{f}_{-(\mathbf{u},\mathbf{t})}, \mathbf{r}, \boldsymbol{w}|\alpha, \beta) = \\ &= P(\mathbf{f}, \mathbf{r}, \boldsymbol{w}|\alpha, \beta), \end{aligned} \tag{7.14}$$

and the denominator from 7.13, which does not depend on $r_{f,s}$

$$\begin{aligned} &\frac{P(r_{f,s}, \mathbf{r}_{-(\mathbf{f},\mathbf{s})}, \mathbf{f}, \boldsymbol{w}|\alpha, \beta)}{P(\mathbf{r}_{-(\mathbf{f},\mathbf{s})}, \mathbf{f}, \boldsymbol{w}|\alpha, \beta)} \propto P(r_{f,s}, \mathbf{r}_{-(\mathbf{f},\mathbf{s})}, \mathbf{f}, \boldsymbol{w}|\alpha, \beta) = \\ &= P(\mathbf{f}, \mathbf{r}, \boldsymbol{w}|\alpha, \beta). \end{aligned} \tag{7.15}$$

Note that $f_{u,t} \cup \mathbf{f}_{-(\mathbf{u},\mathbf{t})} = \mathbf{f}$ and $r_{f,s} \cup \mathbf{r}_{-(\mathbf{f},\mathbf{s})} = \mathbf{r}$. It is also worth mentioning that the conditional probabilities for $f_{u,t}$ and $r_{f,s}$ are proportional to the same formula so far. That is why we will continue

with deriving them together. The derivation will be split into two parts when necessarily.

Let us start with expressing the joint probability for all of the random variables in the LDA-Frames model except of φ and θ that are required to be integrated out in the collapsed Gibbs sampling

$$P(\mathbf{f}, \mathbf{r}, \boldsymbol{w}|\alpha, \beta) = \int_{\varphi} \int_{\theta} P(\mathbf{f}, \mathbf{r}, \boldsymbol{w}, \varphi, \theta|\alpha, \beta)\, d\varphi\, d\theta. \tag{7.16}$$

Now we can expand the probability into the product of factors based on the graphical model from figure 7.1

$$\int_{\varphi} \int_{\theta} P(\varphi|\alpha) P(\mathbf{f}|\varphi) P(\theta|\beta) P(\boldsymbol{w}|\mathbf{f}, \mathbf{r}, \theta)\, d\varphi\, d\theta. \tag{7.17}$$

Because $\int\int x \times y\, dx\, dy = (\int x\, dx) \times (\int y\, dy)$, we can separate the integrals based on the terms being integrated

$$\int_{\varphi} P(\varphi|\alpha) P(\mathbf{f}|\varphi)\, d\varphi \times \int_{\theta} P(\theta|\beta) P(\boldsymbol{w}|\mathbf{f}, \mathbf{r}, \theta)\, d\theta. \tag{7.18}$$

Let us denote

$$\int_{\varphi} P(\varphi|\alpha) P(\mathbf{f}|\varphi)\, d\varphi \equiv A \qquad \int_{\theta} P(\theta|\beta) P(\boldsymbol{w}|\mathbf{f}, \mathbf{r}, \theta)\, d\theta \equiv B. \tag{7.19}$$

Next, let $fc_{f,u}$ be the number of times frame f is assigned to lexical unit u, $wc_{v,r}$ be the number of times word v is assigned to semantic role r in any frame, and $wc_{*,r}$ be the value of $wc_{v,r}$ summed over all words from the vocabulary. Formally,

$$fc_{x,u} = \sum_{t=1}^{T_u} I(f_{u,t} = x) \tag{7.20}$$

$$wc_{v,x} = \sum_{u=1}^{U} \sum_{t=1}^{T_u} \sum_{s=1}^{S} I(r_{f_{u,t},s} = x \,\&\, v = w_{u,t,s}) \tag{7.21}$$

$$wc_{*,x} = \sum_{u=1}^{U} \sum_{t=1}^{T_u} \sum_{s=1}^{S} I(r_{f_{u,t},s} = x). \tag{7.22}$$

Further, we will derive the formula for A and B separately. Let us expand the A term into the products of individual lexical units

$$\begin{aligned} A &= \int_{\varphi} \prod_{u=1}^{U} P(\varphi_u|\alpha)\, P(\mathbf{f_u}|\varphi_u)\, d\varphi = \\ &= \prod_{u=1}^{U} \int_{\varphi_u} P(\varphi_u|\alpha)\, P(\mathbf{f_u}|\varphi_u)\, d\varphi_u = \end{aligned} \tag{7.23}$$

and further expand out the Dirichlet prior and the multinomial distribution according to their definitions.

$$= \prod_{u=1}^{U} \int_{\varphi_u} \frac{\Gamma(\sum_{i=1}^{F} \alpha_i)}{\prod_{i=1}^{F} \Gamma(\alpha_i)} \prod_{i=1}^{F} \varphi_{u,i}^{\alpha_i - 1} \prod_{t=1}^{T_u} \varphi_{u,f_{u,t}} \, d\varphi_u = \tag{7.24}$$

$$= \prod_{u=1}^{U} \int_{\varphi_u} \frac{\Gamma(\sum_{i=1}^{F} \alpha_i)}{\prod_{i=1}^{F} \Gamma(\alpha_i)} \prod_{i=1}^{F} \varphi_{u,i}^{\alpha_i - 1} \prod_{i=1}^{F} \varphi_{u,i}^{fc_{i,u}} \, d\varphi_u = \tag{7.25}$$

Because the Dirichlet distribution is a conjugate prior for the multinomial distribution, we can now merge the products together

$$= \prod_{u=1}^{U} \int_{\varphi_u} \frac{\Gamma(\sum_{i=1}^{F} \alpha_i)}{\prod_{i=1}^{F} \Gamma(\alpha_i)} \prod_{i=1}^{F} \varphi_{u,i}^{\alpha_i + fc_{i,u} - 1} \, d\varphi_u = \tag{7.26}$$

and multiply the formula by a constant fraction equal to one.

$$\begin{aligned} &= \prod_{u=1}^{U} \frac{\Gamma(\sum_{i=1}^{F} \alpha_i)}{\prod_{i=1}^{F} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{F} \Gamma(fc_{i,u} + \alpha_i)}{\Gamma(\sum_{i=1}^{F} fc_{i,u} + \alpha_i)} \times \\ &\int_{\varphi_u} \frac{\Gamma(\sum_{i=1}^{F} fc_{i,u} + \alpha_i)}{\prod_{i=1}^{F} \Gamma(fc_{i,u} + \alpha_i)} \prod_{i=1}^{F} \varphi_{u,i}^{\alpha_i + fc_{i,u} - 1} \, d\varphi_u = \end{aligned} \tag{7.27}$$

Note now that the Dirichlet distribution under the integral is integrated over its entire support density, and thus evaluates to 1.

$$= \prod_{u=1}^{U} \frac{\Gamma(\sum_{i=1}^{F} \alpha_i)}{\prod_{i=1}^{F} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{F} \Gamma(fc_{i,u} + \alpha_i)}{\Gamma(\sum_{i=1}^{F} fc_{i,u} + \alpha_i)} \propto \tag{7.28}$$

Finally, we can drop out the remaining gamma function fraction, which only depends on constant hyperparameter α

$$\propto \prod_{u=1}^{U} \frac{\prod_{i=1}^{F} \Gamma(fc_{i,u} + \alpha_i)}{\Gamma(\sum_{i=1}^{F} fc_{i,u} + \alpha_i)}. \tag{7.29}$$

Now, we will repeat the same procedure for formula B:

$$\begin{aligned} B &= \int_{\theta} \prod_{i=1}^{R} P(\theta_i|\beta) \prod_{u=1}^{U} \prod_{t=1}^{T_u} \prod_{s=1}^{S} P(w_{u,t,s}|\theta_{r_{f_{u,t},s}}) \, d\theta = \\ &= \prod_{i=1}^{R} \int_{\theta_i} P(\theta_i|\beta) \prod_{u=1}^{U} \prod_{t=1}^{T_u} \prod_{s=1}^{S} P(w_{u,t,s}|\theta_{r_{f_{u,t},s}}) \, d\theta_i = \end{aligned} \tag{7.30}$$

$$= \prod_{i=1}^{R} \int_{\theta_i} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\sum_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \theta_{i,v}^{\beta_v - 1} \prod_{u=1}^{U} \prod_{t=1}^{T_u} \prod_{s=1}^{S} \theta_{r_{f_{u,t},s}, w_{u,t,s}} \, d\theta_i = \tag{7.31}$$

$$= \prod_{i=1}^{R} \int_{\theta_i} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\sum_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \theta_{i,v}^{\beta_v - 1} \prod_{v=1}^{V} \theta_{i,v}^{wc_{v,i}} \, d\theta_i = \tag{7.32}$$

$$= \prod_{i=1}^{R} \int_{\theta_i} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\sum_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \theta_{i,v}^{\beta_v + wc_{v,i} - 1} \, d\theta_i = \tag{7.33}$$

$$\begin{aligned} &= \prod_{i=1}^{R} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\sum_{v=1}^{V} \Gamma(\beta_v)} \frac{\prod_{v=1}^{V} \Gamma(wc_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc_{v,i} + \beta_v)} \times \\ &\int_{\theta_i} \frac{\Gamma(\sum_{v=1}^{V} wc_{v,i} + \beta_v)}{\prod_{v=1}^{V} \Gamma(wc_{v,i} + \beta_v)} \prod_{v=1}^{V} \theta_{i,v}^{\beta_v + wc_{v,i} - 1} \, d\theta_i = \end{aligned} \tag{7.34}$$

$$= \prod_{i=1}^{R} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\sum_{v=1}^{V} \Gamma(\beta_v)} \frac{\prod_{v=1}^{V} \Gamma(wc_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc_{v,i} + \beta_v)} \propto \tag{7.35}$$

$$\propto \prod_{i=1}^{R} \frac{\prod_{v=1}^{V} \Gamma(wc_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc_{v,i} + \beta_v)}. \tag{7.36}$$

The joint probability of the collapsed model is thus given by the product

$$A \times B = \prod_{u=1}^{U} \frac{\prod_{i=1}^{F} \Gamma(fc_{i,u} + \alpha_i)}{\Gamma(\sum_{i=1}^{F} fc_{i,u} + \alpha_i)} \times \prod_{i=1}^{R} \frac{\prod_{v=1}^{V} \Gamma(wc_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc_{v,i} + \beta_v)}. \tag{7.37}$$

The probabilities for sampling frames and roles will be derived separately.

7.2.1 *Sampling Frames*

Supposing we are sampling a frame at position $u = a$, $t = b$. We can separate the terms dependent on the current position (a, b),

$$\begin{aligned} &P(f_{a,b} | \mathbf{f}_{-(a,b)}, \mathbf{r}, \boldsymbol{w}, \alpha, \beta) = \\ &\frac{\prod_{i=1}^{F} \Gamma(fc_{i,a} + \alpha_i)}{\Gamma(\sum_{i=1}^{F} fc_{i,a} + \alpha_i)} \times \prod_{u \neq a} \frac{\prod_{i=1}^{F} \Gamma(fc_{i,u} + \alpha_i)}{\Gamma(\sum_{i=1}^{F} fc_{i,u} + \alpha_i)} \times \\ &\prod_{i=1}^{R} \frac{\prod_{v=w_{a,b,*}} \Gamma(wc_{v,i} + \beta_v) \times \prod_{v \neq w_{a,b,*}} \Gamma(wc_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc_{v,i} + \beta_v)} \propto \end{aligned} \tag{7.38}$$

where $w_{a,b,*}$ denotes words at any slot of the position (a, b), and then drop out terms that do not depend on (a, b).

$$\propto \frac{\prod_{i=1}^{F} \Gamma(fc_{i,a} + \alpha_i)}{\Gamma(\sum_{i=1}^{F} fc_{i,a} + \alpha_i)} \times \prod_{i=1}^{R} \frac{\prod_{v=w_{a,b,*}} \Gamma(wc_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc_{v,i} + \beta_v)} \propto \tag{7.39}$$

Let $fc^{-(a,b)}_{x,u}$ be defined the same way as $fc_{x,u}$ without the counts for the position (a,b). It is straightforward that for counts that do not include the position (a,b), $fc^{-(a,b)}_{x,u} = fc_{x,u}$, and for counts that do include the position (a,b), $fc^{-(a,b)}_{x,u} = fc_{x,u} + 1$. Similarly, let $wc^{-(a,b,c)}_{v,x}$ be defined the same way as $wc_{v,x}$ without the counts for the position (a,b,c), where c corresponds to a slot number. Then for counts that do not include position (a,b,c), $wc^{-(a,b,c)}_{v,x} = wc_{v,x}$, and for counts that do, $wc^{-(a,b,c)}_{v,x} = wc_{v,x} + 1$. The term $wc^{-(a,b,*)}_{v,i}$ denotes the same count as $wc^{-(a,b,c)}_{v,x}$, but it excludes counts for all frame slots. The constant $C_{f_{a,b},i}$ denotes the number of repetitions of role i in any slot of frame $f_{a,b}$.

$$\begin{aligned}
&\propto \frac{\Gamma(fc^{-(a,b)}_{f_{a,b},a} + \alpha_{f_{a,b}} + 1) \times \prod_{i \neq f_{a,b}} \Gamma(fc^{-(a,b)}_{i,a} + \alpha_i)}{\Gamma(1 + \sum_{i=1}^{F} fc^{-(a,b)}_{i,a} + \alpha_i)} \times \\
&\prod_{i = r_{f_{a,b},*}} \frac{\prod_{v = w_{a,b,*}} \Gamma(wc^{-(a,b,*)}_{v,i} + \beta_v + C_{f_{a,b},i})}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b,*)}_{v,i} + \beta_v + C_{f_{a,b},i})} \times \\
&\prod_{i \neq r_{f_{a,b},*}} \frac{\prod_{v = w_{a,b,*}} \Gamma(wc^{-(a,b,*)}_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b,*)}_{v,i} + \beta_v)} =
\end{aligned} \tag{7.40}$$

Using the fact that $\Gamma(x+1) = x \times \Gamma(x)$, we can expand the terms dependent on the current position

$$\begin{aligned}
&= \frac{fc^{-(a,b)}_{f_{a,b},a} + \alpha_{f_{a,b}}}{\sum_{i=1}^{F} fc^{-(a,b)}_{i,a} + \alpha_i} \times \\
&\frac{\Gamma(fc^{-(a,b)}_{f_{a,b},a} + \alpha_{f_{a,b}}) \times \prod_{i \neq f_{a,b}} \Gamma(fc^{-(a,b)}_{i,a} + \alpha_i)}{\Gamma(\sum_{i=1}^{F} fc^{-(a,b)}_{i,a} + \alpha_i)} \times \\
&\prod_{s=1}^{S} \frac{wc^{-(a,b,s)}_{w_{a,b,s}, r_{f_{a,b},s}} + \beta_{w_{a,b,s}}}{\sum_{v=1}^{V} wc^{-(a,b,s)}_{v, r_{f_{a,b},s}} + \beta_v} \\
&\prod_{i = r_{f_{a,b},*}} \frac{\prod_{v = w_{a,b,*}} \Gamma(wc^{-(a,b,*)}_{v,i} + \beta_v}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b,*)}_{v,i} + \beta_v} \times \\
&\prod_{i \neq r_{f_{a,b},*}} \frac{\prod_{v = w_{a,b,*}} \Gamma(wc^{-(a,b,*)}_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b,*)}_{v,i} + \beta_v)} =
\end{aligned} \tag{7.41}$$

and then group back the products of gamma functions

$$\begin{aligned}
&= \frac{fc^{-(a,b)}_{f_{a,b},a} + \alpha_{f_{a,b}}}{\sum_{i=1}^{F} fc^{-(a,b)}_{i,a} + \alpha_i} \times \frac{\prod_{i=1}^{F} \Gamma(fc^{-(a,b)}_{i,a} + \alpha_i)}{\Gamma(\sum_{i=1}^{F} fc^{-(a,b)}_{i,a} + \alpha_i)} \times \\
&\prod_{s=1}^{S} \frac{wc^{-(a,b,s)}_{w_{a,b,s}, r_{f_{a,b},s}} + \beta_{w_{a,b,s}}}{\sum_{v=1}^{V} wc^{-(a,b,s)}_{v, r_{f_{a,b},s}} + \beta_v} \prod_{i=1}^{R} \frac{\prod_{v = w_{a,b,*}} \Gamma(wc^{-(a,b,*)}_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b,*)}_{v,i} + \beta_v)} \propto
\end{aligned}$$

$$(7.42)$$

now we can remove terms that do not depend on the current position and get the final expression

$$\propto (fc^{-(a,b)}_{f_{a,b},a} + \alpha_{f_{a,b}}) \times \prod_{s=1}^{S} \frac{wc^{-(a,b,s)}_{w_{a,b,s},r_{f_{a,b},s}} + \beta_{w_{a,b,s}}}{\sum_{v=1}^{V} wc^{-(a,b,s)}_{v,r_{f_{a,b},s}} + \beta_v}. \tag{7.43}$$

7.2.2 *Sampling Roles*

Supposing we are sampling a role at position $f = a$, $s = b$. Let $wc^{-(a,b)}_{v,x}$ express the same value as $wc_{v,x}$ without the counts for frame a and slot b, and let $C_{f,s,v}$ denote the number of times word v occurs as a realization of slot s of frame f. The derivation of the formula for sampling roles is very similar to the derivation of the formula for sampling frames:

$$P(r_{a,b}|\mathbf{f}, \mathbf{r}_{-(a,b)}, \mathbf{w}, \alpha, \beta) \propto \prod_{i=1}^{R} \frac{\prod_{v=1}^{V} \Gamma(wc_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc_{v,i} + \beta_v)} = \tag{7.44}$$

$$= \frac{\prod_{v=1}^{V} \Gamma(wc^{-(a,b)}_{v,r_{a,b}} + \beta_v + C_{a,b,v})}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b)}_{v,r_{a,b}} + \beta_v + C_{a,b,v})} \times \prod_{i \neq r_{a,b}} \frac{\prod_{v=1}^{V} \Gamma(wc^{-(a,b)}_{v,r_{a,b}} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b)}_{v,r_{a,b}} + \beta_v)} \propto \tag{7.45}$$

$$\propto \left(\frac{\prod_{v=1}^{V} wc^{-(a,b)}_{v,r_{a,b}} + \beta_v}{\sum_{v=1}^{V} wc^{-(a,b)}_{v,r_{a,b}} + \beta_v} \right)^{C_{a,b,v}} \times \frac{\prod_{v=1}^{V} \Gamma(wc^{-(a,b)}_{v,r_{a,b}} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b)}_{v,r_{a,b}} + \beta_v)} \times \prod_{i \neq r_{a,b}} \frac{\prod_{v=1}^{V} \Gamma(wc^{-(a,b)}_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b)}_{v,i} + \beta_v)} = \tag{7.46}$$

$$= \left(\frac{\prod_{v=1}^{V} wc^{-(a,b)}_{v,r_{a,b}} + \beta_v}{\sum_{v=1}^{V} wc^{-(a,b)}_{v,r_{a,b}} + \beta_v} \right)^{C_{a,b,v}} \times \prod_{i=1}^{R} \frac{\prod_{v=1}^{V} \Gamma(wc^{-(a,b)}_{v,i} + \beta_v)}{\Gamma(\sum_{v=1}^{V} wc^{-(a,b)}_{v,i} + \beta_v)} \propto \tag{7.47}$$

$$\propto \left(\frac{\prod_{v=1}^{V} wc^{-(a,b)}_{v,r_{a,b}} + \beta_v}{\sum_{v=1}^{V} wc^{-(a,b)}_{v,r_{a,b}} + \beta_v} \right)^{C_{a,b,v}}. \tag{7.48}$$

If the hyperparameters α_i and β_i are all the same value (which is sufficient for most use cases), i.e. $\alpha = \alpha_1 = \alpha_2 = \ldots \alpha_F$, and $\beta = \beta_1 = \beta_2 = \ldots \beta_V$, we can use a shorthand notation for the resulting equations 7.43 and 7.48:

$$P(f_{u,t}|\mathbf{f}_{-(u,t)}, \mathbf{r}, \boldsymbol{w}, \alpha, \beta) \propto (fc^{-(u,t)}_{f_{u,t},u} + \alpha) \times \prod_{s=1}^{S} \frac{wc^{-(u,t,s)}_{w_{u,t,s},r_{f_{u,t},s}} + \beta}{wc^{-(u,t,s)}_{*,r_{f_{u,t},s}} + V\beta} \tag{7.49}$$

$$P(r_{f,s}|\mathbf{f}, \mathbf{r}_{-(f,s)}, \boldsymbol{w}, \alpha, \beta) \propto \prod_{v=1}^{V} \left(\frac{wc^{-(f,s)}_{v,r_{f,s}} + \beta}{wc^{-(f,s)}_{*,r_{f,s}} + V\beta} \right)^{C_{f,s,v}} . \tag{7.50}$$

Because the expressions were taken up to proportionality, all what remains is to normalize them in order to get probabilities:

$$P(f_{u,t}|\mathbf{f}_{-(u,t)}, \mathbf{r}, \boldsymbol{w}, \alpha, \beta) = \frac{(fc^{-(u,t)}_{f_{u,t},u} + \alpha) \times \prod_{s=1}^{S} \frac{wc^{-(u,t,s)}_{w_{u,t,s},r_{f_{u,t},s}} + \beta}{wc^{-(u,t,s)}_{*,r_{f_{u,t},s}} + V\beta}}{\sum_{i=1}^{F} \left[(fc^{-(u,t)}_{i,u} + \alpha) \times \prod_{s=1}^{S} \frac{wc^{-(u,t,s)}_{w_{u,t,s},r_{i,s}} + \beta}{wc^{-(u,t,s)}_{*,r_{i,s}} + V\beta} \right]} \tag{7.51}$$

$$P(r_{f,s}|\mathbf{f}, \mathbf{r}_{-(f,s)}, \boldsymbol{w}, \alpha, \beta) = \frac{\prod_{v=1}^{V} \left(\frac{wc^{-(f,s)}_{v,r_{f,s}} + \beta}{wc^{-(f,s)}_{*,r_{f,s}} + V\beta} \right)^{C_{f,s,v}}}{\sum_{i=1}^{R} \left[\prod_{v=1}^{V} \left(\frac{wc^{-(f,s)}_{v,r_{f,s}} + \beta}{wc^{-(f,s)}_{*,r_{f,s}} + V\beta} \right)^{C_{f,s,v}} \right]} . \tag{7.52}$$

Sampling from the categorical distribution over frames is now straightforward and is described in algorithm 7.2.

Algorithm 7.2 Sampling from the categorical distribution over frames.

```
Choose t from Uniform(0, 1)
s ← 0
for i ∈ {1, 2, ..., F} do
    s ← s + P(f_{u,t} = i | f_{-(u,t)}, r, w, α, β)
    if t < s then
        return i
    end if
end for
```

The algorithm starts with sampling a real-valued threshold t between 0 and 1 from the uniform distribution. Then it iterates through all frame numbers and adds their probability to the variable s, previously initialized to 0. If the sum stored in s exceeds the threshold t, the algorithm returns the current frame number. The sampler for the categorical distribution over roles can be constructed analogously.

8 NON-PARAMETRIC LDA-FRAMES

As was stated in the previous chapter, the LDA-Frames model is required to define the number of frames and roles in advance. It is not clear, however, how to select the best values which are depending on several factors. First of all, the number of frames and roles usually increase with growing size of the training corpus. If the training data is small and covers only a small proportion of lexical unit usage patterns, the number of semantic frames should be small as well. The parameters are also affected by the granularity of roles and frames. Apart from selecting the parameters based on an annotated data set (which we are trying to avoid), a good way of automatic estimation of parameters is to select those that maximize the posterior probability of the model given training data.

The LDA-frames algorithm generates frames from the Dirichlet distribution, which requires a fixed number of components. Similarly, latent variables $r_{f,s}$ are chosen from a fixed set of semantic roles. In order to be able to update the number of frames and roles during the inference process, we need to use stochastic processes that assign probability distributions to possibly infinite structures instead of using Dirichlet distributions over a finite set of components. Specifically, we will add the **Chinese restaurant process** prior for the $r_{f,s}$ variables, and replace the Dirichlet distribution from which the semantic frames are generated with the **Dirichlet process**. Most of the ideas presented in this chapter are based on my previous work published in Materna (2013).

Chinese restaurant process

Dirichlet process

8.1 DIRICHLET PROCESS

The Dirichlet distribution can be seen as a bag full of k-sided dice with various probabilities of getting the k states after rolling. When we sample from the Dirichlet distribution, we pick one die from the bag. Sampling from the corresponding multinomial distribution in this analogy is simulated by rolling the picked die. Now let us move on to the processes that enables modeling a distribution over an infinite set of discrete components.

The Dirichlet process (Ferguson, 1973) is a generalization of the Dirichlet distribution, which enables working with an infinite set of events. In the dice analogy, it means that the dice have an infinite number of sides. The sampling space, however, remains discrete.

We will formally introduce the Dirichlet process as a generalization of the Dirichlet distribution. Let us have a look at the Dirichlet

distribution's parameter $\boldsymbol{\alpha}$ as a categorical distribution G_0 over K categories scaled by a scalar parameter α_0:

$$\mathrm{Dir}(\boldsymbol{\alpha}) \equiv \mathrm{Dir}(\alpha_0 G_0). \tag{8.1}$$

The distribution G_0 can be expressed using the Dirac delta function

$$G_0 = \sum_{k=1}^{K} \pi_{0k} \delta(x - k), \tag{8.2}$$

where π_{0k} are non-negative real-valued weights such that

$$\sum_{k=1}^{K} \pi_{0k} = 1, \tag{8.3}$$

and $\delta(x)$ is the Dirac function, which is defined for all real numbers so that

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{else.} \end{cases} \tag{8.4}$$

base distribution

The distribution G_0 is usually called the **base distribution** of the Dirichlet distribution $\mathrm{Dir}(\alpha_0 G_0)$. It can easily be seen that the base distribution has K spikes at locations $1, 2, \ldots, K$ with heights $\pi_{01}, \pi_{02}, \ldots, \pi_{0K}$, and the samples from $\mathrm{Dir}(\alpha_0 G_0)$ have the same form as G_0

$$G \sim \mathrm{Dir}(\alpha_0 G_0) \qquad G = \sum_{k=1}^{K} \pi_k \delta(x - k), \tag{8.5}$$

where

$$\sum_{k=1}^{K} \pi_k = 1. \tag{8.6}$$

The Dirichlet process extends the Dirichlet distribution with considering any (with some restrictions) infinite distribution H_0 defined on a support set $S(H_0)$ instead of the categorical base distribution G_0:

$$H \sim \mathrm{DP}(\alpha_0, H_0) \qquad H = \sum_{k=1}^{\infty} \pi_k \delta(x - \phi_k), \tag{8.7}$$

where

$$\sum_{k=1}^{\infty} \pi_k = 1 \qquad \phi_k \in S(H_0). \tag{8.8}$$

The samples $H \sim \mathrm{DP}(\alpha_0, H_0)$ are still discrete, but instead of category labels $k \in \{1, 2, \ldots, K\}$ as is the case of the Dirichlet

distribution, the point masses of the Dirichlet process lie somewhere on the support set $S(H_0)$.

One of the nice things about the Dirichlet process is that we can aggregate the components of a Dirichlet process and get a Dirichlet distribution. Specifically, any finite partition of the support set $S = \{S_1, S_2, \ldots, S_J\}$ of a Dirichlet process $DP(\alpha_0, H_0)$ defines a Dirichlet distribution such that the sums of the weights π_k in each of these J subsets are distributed according to the Dirichlet distribution, parametrized by α_0 and the base distribution G_0, whose weights are equal to the integrals of the base distribution H_0 over the subsets $S_1, S_2, \ldots, S_J$ (Frigyik et al., 2010). Let us denote

$$H(S_i) = \sum_{k;\, \phi_k \in S_i} \pi_k, \tag{8.9}$$

then

$$\begin{aligned} &(H(S_1), H(S_2), \ldots H(S_J)) \sim \\ &\mathrm{Dir}(\alpha_0 H_0(S_1), \alpha_0 H_0(S_2), \ldots, \alpha_0 H_0(S_J)). \end{aligned} \tag{8.10}$$

8.2 CHINESE RESTAURANT PROCESS AND STICK BREAKING PROCESS

An important and natural question about Dirichlet processes is how to generate samples from them. In this section, we will discuss two methods important for this work – the Chinese restaurant process and the Stick breaking process.

The Chinese restaurant process (Aldous, 1985), sometimes called the Pólya Urn sequence, is a sampling scheme that uses a metaphor with a Chinese restaurant. Consider a restaurant with an infinite number of tables, each of them associated with a dish ϕ_k, and N customers choosing a table. The first customer sits at the first table. The ith customer sits at table k drawn from distribution

$$\begin{aligned} P(\theta_i = \phi_k | \theta_1, \theta_2, \ldots, \theta_{i-1}, \alpha_0) &= \frac{m_k}{\alpha_0 + i - 1} \\ P(\theta_i = \phi_{K+1} | \theta_1, \theta_2, \ldots, \theta_{i-1}, \alpha_0) &= \frac{\alpha_0}{\alpha_0 + i - 1} \end{aligned} \tag{8.11}$$

where K is the number of already occupied tables, m_k is the number of customers sitting at table $k \in \{1, 2, \ldots, K\}$, $\alpha_0 > 0$ is the concentration parameter which controls how often the customer chooses a new table $K + 1$, and

$$\phi_j \sim H_0, \qquad j \in \{1, 2, \ldots, \infty\}. \tag{8.12}$$

The seating plan makes a partition of the customers, where each customer i is associated with a parameter vector θ_i according to the table which he sits. In other words, with seating the customers,

we generate samples from the Dirichlet process with concentration parameter α_0 and the base distribution H_0. The model is illustrated in figure 8.1. The circles along with parameter vectors ϕ_k represent tables, and variables θ_i, placed around tables, represent the dish choice of a customer i.

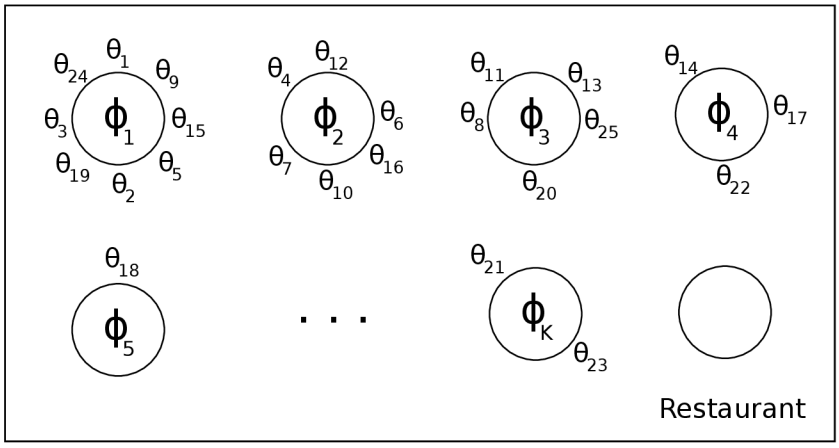


Figure 8.1: Chinese restaurant process.

The Pólya urn sampling scheme (Johnson and Kotz, 1977) for sampling from a Dirichlet process is a generalization of the Pólya urn for the Dirichlet Distribution, which can also be described by using a metaphor. Let an urn be filled with balls of unit size, each colored with one of K colors, where their quantity equals to weights in the scaled base distribution $\alpha_0 G_0$. For non-integer weights, we allow ball fractions. The sampling is realized by choosing a ball from the urn and observing its color. After that, the ball is returned back to the urn together with another unit ball with the same color. Then, the next sample can be taken.

The Pólya urn sampling for the Dirichlet process is analogical, but, instead of a set of balls with K colors, the urn is filled with a liquid that has infinitely many different color mixtures, distributed according to the base distribution H_0 and volume α_0. The first sample in this generalized sampling schema draws a volume of liquid and returns the liquid together with a unit ball of the same color back to the urn. In the subsequent turns, we either draw a ball or a sample of the liquid.

A related method to the Chinese restaurant process and the Pólya urn is the Stick-breaking construction. The stick breaking is an approach for generating a random vector with a $\mathrm{Dir}(\alpha_0)$ distribution. It involves iteratively breaking a stick of a unit length into K pieces in such a way that the lengths of the pieces follow the Dirichlet distribution. When considering an unbound variant of the stick-breaking process, the number of pieces K is equal to ∞.

The process is formally realized using independent draws from the beta distribution. Let the interval $(0, 1)$ represent a unit-length stick. The process starts with drawing a value v_1 from the $\text{Beta}(a, b)$ distribution. Let $\pi_1 = v_1$ denote the first piece of the stick and $1 - \pi_1$ denotes the rest of the stick. The procedure continues recursively, producing new pieces $\pi_2 = v_2(1 - \pi_1), \pi_3 = v_3(1 - \pi_1 - \pi_2)$, etc. The length of any fragment π_i of the stick is defined as

$$\pi_i = v_i \prod_{j=1}^{i-1} (1 - v_j). \tag{8.13}$$

Sethuraman (1994) showed that the resulting sequence satisfies

$$\sum_{i=1}^{\infty} \pi_i = 1. \tag{8.14}$$

An important special case of the stick-breaking process is constructed when we set $a = 1$. The one-parameter stochastic process for $a = 1$ and $\alpha_0 = b$ is known as the **GEM distribution** (Pitman, 2002) denoted *GEM distribution*

$$\boldsymbol{\pi} \sim \text{GEM}(\alpha_0). \tag{8.15}$$

The relation between the GEM distribution, the Chinese restaurant process, and the Dirichlet process is very close. Let $z_1, z_2, \ldots, z_N$ be a sequence of integer random variables sampled from the categorical distribution parametrized with $\boldsymbol{\pi}$, i.e.

$$P(z_j = i | \boldsymbol{\pi}) = \pi_i. \tag{8.16}$$

The sampled values define a random partition on the integers $1, 2, \ldots, N$ such that two integers i, j belong to the same cluster if $z_i = z_j$. Pitman (2002) showed that this distribution on partitions is the same as the distribution on partitions induced by the Chinese restaurant process, where the GEM parameter α_0 controls the partitioning in the same way as the parameter α_0 in the Chinese restaurant process.

Let $\{\phi_i\}$ be an infinite sequence of draws from a base distribution H_0 defined on a support set $S(H_0)$

$$\phi_i \in S(H_0), \tag{8.17}$$

then

$$H = \sum_{k=1}^{\infty} \pi_k \delta(x - \phi_k), \tag{8.18}$$

where δ is the Dirac function and $\boldsymbol{\pi} \sim \text{GEM}(\alpha_0)$, defines a Dirichlet process with the base distribution H_0 and the concentration parameter α_0 (Sethuraman, 1994).

8.3 ESTIMATING THE NUMBER OF ROLES

The parameter estimators will be derived separately. Let us start with the number of roles, which is much easier to estimate than the number of frames. The LDA-Frames graphical model in figure 7.1 does not place any prior distribution on the role latent variables $\mathbf{r}$. A prior is not essential there, because the variables are shared and their number is fixed. In the non-parametric model, however, it may be useful for the role variables to use a prior, which controls their values and allows to create new roles if necessary.

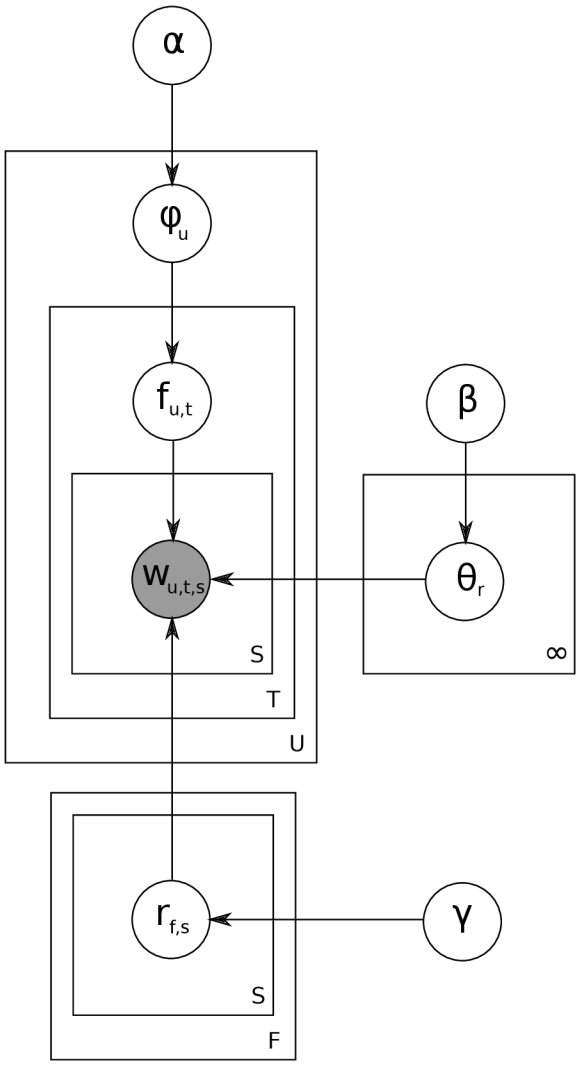


Figure 8.2: Estimating the number of roles in LDA-frames.

An appropriate prior with clustering behavior that can be used to model the creation of new roles is the Chinese restaurant process. Its incorporation into the LDA-Frames model is visualized in the graphical model in figure 8.2. The semantic roles $\mathbf{r}$ are there generated

from the Chinese restaurant process with parameter γ, and the base distribution $\mathrm{Dir}(\beta)$, so that

$$\theta_r \sim \mathrm{Dir}(\beta). \tag{8.19}$$

Notice that the number of roles in the right-most plate changed from R to ∞, which means that the number is theoretically unbounded.

In deriving the equation expressing the conditional probabilities for sampling roles, we can start with the expression for the parametric LDA-Frames and extend it with another term, representing the Chinese restaurant process

$$\begin{aligned} &P(r_{f,s}|\mathbf{f}, \mathbf{r}_{-(f,s)}, \boldsymbol{w}, \alpha, \beta, \gamma) \propto \\ &\mathrm{CRP}(\gamma) \times \prod_{v=1}^{V} \left(\frac{wc_{v,r_{f,s}}^{-(f,s)} + \beta}{wc_{*,r_{f,s}}^{-(f,s)} + V\beta} \right)^{C_{f,s,v}}. \end{aligned} \tag{8.20}$$

Now let

$$P(r_{f,s}^{old}|\mathbf{r}_{-(f,s)}, \mathbf{f}, \boldsymbol{w}, \alpha, \beta, \gamma) \tag{8.21}$$

denote the conditional probability of sampling $r_{f,s} \in \{1, 2, \ldots, R\}$, where R is the highest already sampled role identifier, and

$$P(r_{f,s}^{new}|\mathbf{r}_{-(f,s)}, \mathbf{f}, \boldsymbol{w}, \alpha, \beta, \gamma) \tag{8.22}$$

denote the conditional probability of sampling $r_{f,s}$ with a new value. With the usage of the expression from 8.11, the probability of sampling an existing role is given by

$$\begin{aligned} &P(r_{f,s}^{old}|\mathbf{f}, \mathbf{r}_{-(f,s)}, \boldsymbol{w}, \alpha, \beta, \gamma) \propto \\ &\propto \frac{rc_{r_{f,s}}^{-(f,s)}}{\gamma + \sum_{i=1}^{R} rc_i - 1} \prod_{v=1}^{V} \left(\frac{wc_{v,r_{f,s}}^{-(f,s)} + \beta}{wc_{*,r_{f,s}}^{-(f,s)} + V\beta} \right)^{C_{f,s,v}}. \end{aligned} \tag{8.23}$$

where rc_i denotes the number of times role i is used in any slot of any frame, and the expression $rc_i^{-(f,s)}$ is the same with an exclusion of the current position.

Since the denominator of the $\mathrm{CRP}(\gamma)$ term is constant for all values of $r_{f,s}^{old}$, we get

$$\begin{aligned} &P(r_{f,s}^{old}|\mathbf{f}, \mathbf{r}_{-(f,s)}, \boldsymbol{w}, \alpha, \beta, \gamma) \propto \\ &\propto rc_{r_{f,s}}^{-(f,s)} \prod_{v=1}^{V} \left(\frac{wc_{v,r_{f,s}}^{-(f,s)} + \beta}{wc_{*,r_{f,s}}^{-(f,s)} + V\beta} \right)^{C_{f,s,v}}. \end{aligned} \tag{8.24}$$

The probability of sampling a new role $r_{f,s}^{new}$ is from 8.11 and 8.20 given by

$$\begin{aligned} &P(r_{f,s}^{new}|\mathbf{f}, \mathbf{r}_{-(f,s)}, \boldsymbol{w}, \alpha, \beta, \gamma) \propto \\ &\propto \frac{\gamma}{\gamma + \sum_{i=1}^{R} rc_i - 1} \prod_{v=1}^{V} \left(\frac{wc_{v,r_{f,s}}^{-(f,s)} + \beta}{wc_{*,r_{f,s}}^{-(f,s)} + V\beta} \right)^{C_{f,s,v}}. \end{aligned} \tag{8.25}$$

Again, we can remove the denominator, and moreover, we can also remove both occurrences of 'wc' counts, which are equal to 0. This gives us a simplified formula

$$P(r_{f,s}^{new}|\mathbf{f}, \mathbf{r}_{-(f,s)}, \boldsymbol{w}, \alpha, \beta, \gamma) \propto$$
$$\propto \gamma \prod_{v=1}^{V} \frac{1}{V^{C_{f,s,v}}}. \tag{8.26}$$

8.4 ESTIMATING THE NUMBER OF FRAMES

The estimation of the number of frames is more complex than the case of semantic roles. The idea is to use the Dirichlet process instead of the Dirichlet distribution as a prior for φ_u. The question, however, is how to make the sampled frames shared between different lexical units. If we simply replaced the Dirichlet distribution with the Dirichlet process, the model would create unique spikes in the parameter space for each lexical unit, and therefore, the lexical units would most likely have disjoint φ distributions, which would prevent from sharing frames. The solution is to use the **Hierarchical Dirichlet process** (Teh et al., 2006).

Hierarchical Dirichlet process

8.4.1 *Hierarchical Dirichlet Process*

The idea behind the Hierarchical Dirichlet process is to create the base distribution of the Dirichlet process by sampling from another Dirichlet process. In such a hierarchy the samples are drawn from

$$G_j \sim DP(\alpha_0, G_0), \tag{8.27}$$

where the global distribution G_0 is distributed as

$$G_0 \sim DP(\gamma_0, H), \tag{8.28}$$

which is visualized in the graphical model in figure 8.3. The Hierarchical Dirichlet process is used to model the prior of a simple non-parametric mixture model with observed variables $\mathbf{x}$, drawn from a discrete distribution θ. The model can naturally be extended to more than two levels, but it is not necessary for LDA-Frames.

For the further derivations of the sampler for non-parametric LDA-Frames, it is convenient to describe how samples can be drawn from the Hierarchical Dirichlet process. We will start with the stick-breaking construction. Let the global distribution G_0 be expressed as

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta(x - \phi_k), \tag{8.29}$$

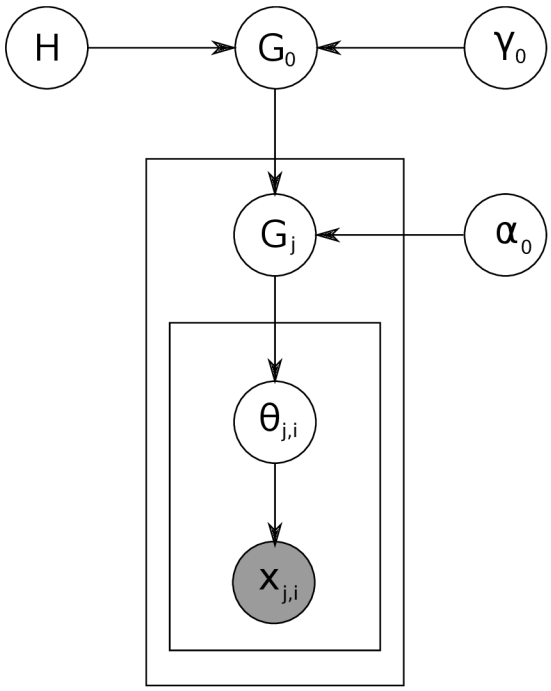


Figure 8.3: Hierarchical Dirichlet process.

where

$$\boldsymbol{\phi} = (\phi_k)_{k=1}^{\infty} \sim H \qquad \boldsymbol{\beta} = (\beta_k)_{k=1}^{\infty} \sim \mathrm{GEM}(\gamma_0). \tag{8.30}$$

Then, since each G_j has support on the same points as G_0, we can write

$$G_J = \sum_{k=1}^{\infty} \pi_{j,k} \delta(x - \phi_k). \tag{8.31}$$

Teh et al. (2006) showed that the weights $\boldsymbol{\pi}_j$ can be computed as

$$\pi_{j,k} = \nu_{j,k} \prod_{l=1}^{k-1} (1 - \nu_{j,l}), \tag{8.32}$$

where

$$\nu_{j,k} \sim \mathrm{Beta}\left(\alpha_0 \beta_k, \alpha_0 \left(1 - \sum_{l=1}^{k} \beta_l\right)\right). \tag{8.33}$$

A parallel with the Chinese restaurant process can be used to describe another method for constructing samples from the Hierarchical Dirichlet process. It is called the **Chinese restaurant franchise**. The metaphor extends the Chinese restaurant process to allow multiple restaurants which share the same set of dishes.

Chinese restaurant franchise

The story is as follows. We have restaurants with a shared menu. A customer comes to a restaurant, sits at the first table, and orders

a dish from the menu. The next customer in the same restaurant can either join the first customer at his table and order the same dish or choose another table and order a dish from the shared menu. Notice that multiple tables in multiple restaurants can serve the same dish.

In this setup, factors $\theta_{j,i}$ correspond to the choice of customer i in restaurant j, dishes in the shared menu correspond to $\{\phi_k\}_{k=1}^{\infty}$ drawn from H, and variables $\psi_{j,t}$ represent the table-specific choice of a dish at table t in restaurant j.

To be able to work with counts, we also introduce variable $m_{j,k}$, representing the number of tables serving dish k in restaurant j, $n_{j,t}$ representing the number of customers in restaurant j sitting at table t, and their aggregate versions

$$m_{*,k} = \sum_j m_{j,k} \qquad m_{j,*} = \sum_k m_{j,k} \qquad m_{*,*} = \sum_j \sum_k m_{j,k}, \tag{8.34}$$

$$n_{*,t} = \sum_j n_{j,t} \qquad n_{j,*} = \sum_t n_{j,t} \qquad n_{*,*} = \sum_j \sum_t n_{j,t}. \tag{8.35}$$

Finally, the relationship between the random variables θ, ϕ, and ψ is stored in variables $t_{j,i}$ and $k_{j,t}$. In particular, $t_{j,i} \in \{1, 2, \ldots, m_{j,*}\}$ represents an identifier of the table in restaurant j, where customer i sits, and $k_{j,t} \in \{1, 2, \ldots, K\}$ represents the menu item, served at table t in restaurant j.

Now we can proceed to a formal description of the generative process. First consider the conditional distribution for $\theta_{j,i}$, where G_j is integrated out

$$P(\theta_{j,i} = \psi_{j,t} | \theta_{j,1}, \theta_{j,2}, \ldots, \theta_{j,i-1}, \alpha_0) = \frac{n_{j,t}}{i + \alpha_0 - 1}, \tag{8.36}$$

$$P(\theta_{j,i} = \psi_{j,m_{j,*}+1} | \theta_{j,1}, \theta_{j,2}, \ldots, \theta_{j,i-1}, \alpha_0) = \frac{\alpha_0}{i + \alpha_0 - 1}. \tag{8.37}$$

If an occupied table is chosen, i.e. $\theta_{j,i} = \psi_{j,t}$, then $t_{j,i} = t$. If an empty table is chosen, we increment $m_{j,*}$ by one, set $\theta_{j,i} = \psi_{j,m_{j,*}}$, $t_{j,i} = m_{j,*}$, and sample $\psi_{j,m_{j,*}}$ according to

$$\psi_{j,m_{j,*}} \sim G_0, \tag{8.38}$$

where G_0 is defined as the top-level Chinese restaurant process with G_0 being integrated out:

$$P(\psi_{j,t} = \phi_k | \psi_{1,1}, \psi_{1,2}, \ldots, \psi_{j,t-1}, \gamma_0) = \frac{m_{*,k}}{m_{*,*} + \gamma_0 - 1}, \tag{8.39}$$

$$P(\psi_{j,t} = \phi_{k+1} | \psi_{1,1}, \psi_{1,2}, \ldots, \psi_{j,t-1}, \gamma_0) = \frac{\gamma_0}{m_{*,*} + \gamma_0 - 1}. \tag{8.40}$$

If an existing menu item is chosen, i.e. $\psi_{j,t} = \phi_k$, we set $k_{j,t} = k$ for the chosen k. If a new menu item is required, we increment K by one, set $\psi_{j,t} = \phi_k$, $k_{j,t} = K$ and sample ϕ_K from the base distribution

$$\phi_K \sim H. \tag{8.41}$$

The Chinese restaurant franchise concept is depicted in figure 8.4. The top rectangle represents a menu from which customers choose their dish. The bottom rectangle group all restaurants that share the same menu.

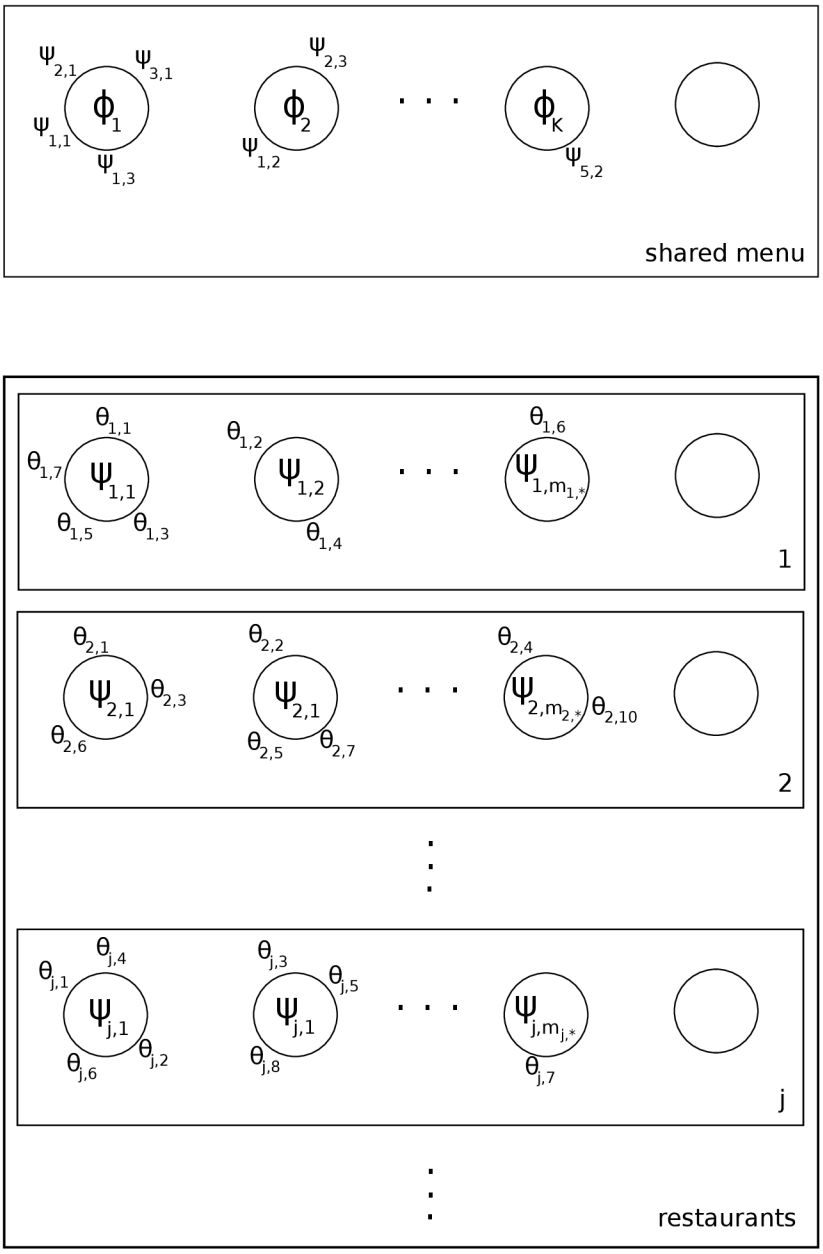


Figure 8.4: Chinese restaurant franchise.

8.4.2 *Fully Non-Parametric Model*

Being equipped with basic knowledge of the Hierarchical Dirichlet process, we can now modify the LDA-Frames model to be able to estimate the number of frames. The point is that we will generate frame variables $f_{u,t}$ from a two-level hierarchy of Dirichlet processes simulated using the Chinese restaurant franchise. The spikes of frames shared among all lexical units (menu items in the Chinese restaurant franchise analogy) are generated from the Dirichlet process χ with concentration parameter δ. Its base distribution is supposed to generate new frames, and is realized by subsequent sampling roles for every slot of a new frame. The lexical-unit-specific Dirichlet processes φ_u use χ as the base distribution, and are governed by the concentration parameter α. The graphical model of the fully non-parametric model is visualized in figure 8.5.

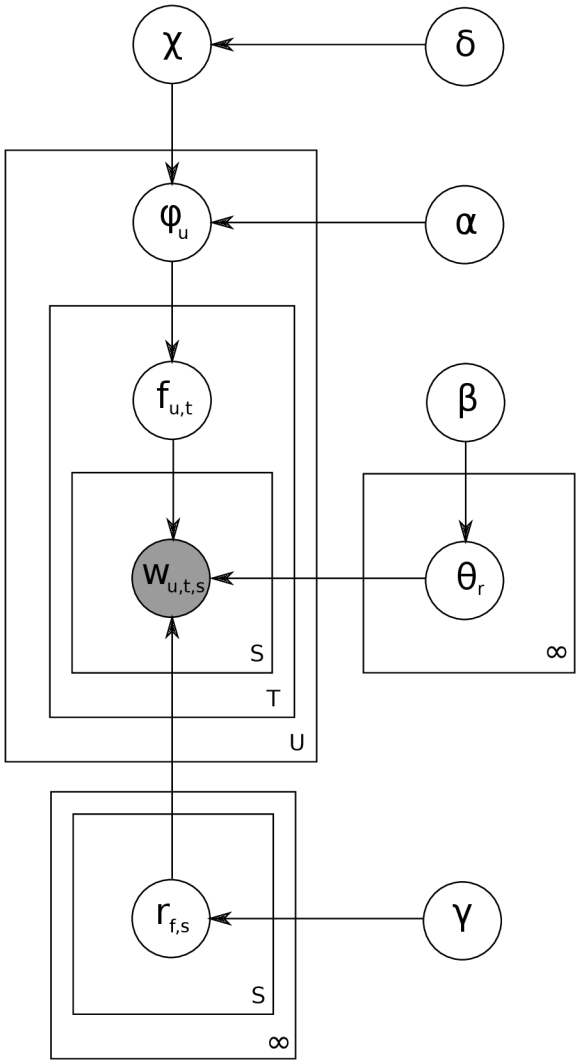


Figure 8.5: Graphical model of Non-parametric LDA-frames.

8.4.3 *Sampling Unbounded Number of Frames*

It only remains to describe formally how to sample frames in the fully non-parametric model, and how to generate samples from the base distribution of χ. We will start with a straightforward Gibbs sampler, sampling variables for all levels of the hierarchy, based on the Chinese restaurant franchise. It will later be merged into a compact model, expressed in a form similar to the parametric version.

According to the Chinese restaurant franchise, a frame identifier $f_{j,i}$ is equal to the menu item which is ordered by customer i in restaurant j,

$$f_{j,i} = k_{j,t_{j,i}}, \tag{8.42}$$

thus, we need to show how to sample $\mathbf{t}$, $\mathbf{k}$, and new frames for $f_{j,i} = K+1$. Let $\mathcal{L}(k)^{-(j,i)}$ denote the proportional likelihood of $f_{j,i} = k$, which is from 7.43 given by

$$\mathcal{L}(k)^{-(j,i)} = \prod_{s=1}^{S} \frac{wc^{-(j,i,s)}_{w_{j,i,s},r_{k,s}} + \beta}{wc^{-(j,i,s)}_{*,r_{k,s}} + V\beta}, \tag{8.43}$$

and let $\mathcal{L}(k^{new})^{-(j,i)}$ denote the proportional likelihood of a new frame, which can be derived from 8.43, by setting the word counts equal to 0:

$$\mathcal{L}(k^{new})^{-(j,i)} = \prod_{s=1}^{S} \frac{0+\beta}{0+V\beta} = \frac{1}{V^S}. \tag{8.44}$$

The conditional distribution of $t_{j,i} = t$, where t is a previously used table is then proportional to

$$P(t_{j,i} = t|\mathbf{t}^{-(j,i)}) \propto n^{-(j,t)}_{j,t} \mathcal{L}(k_{j,i})^{-(j,i)}, \tag{8.45}$$

and the case of an unoccupied table can be calculated by integrating out the possible values of $k_{j,m_{j,*}+1}$ as

$$\begin{aligned} &P(t_{j,i} = m_{j,*}+1|\mathbf{t}^{-(j,i)}, \mathbf{k}) \propto \\ &\alpha \left(\sum_{k=1}^{K} \frac{m_{*,k}}{m_{*,*}+\delta-1} \mathcal{L}(k)^{-(j,i)} + \frac{\delta}{m_{*,*}+\delta-1} \mathcal{L}(k^{new})^{-(j,i)} \right). \end{aligned} \tag{8.46}$$

If the value of $t_{j,i}$ is equal to $m_{j,*}+1$, i.e. a new table is selected, we obtain a sample of $k_{j,t_{m_{j,*}+1}}$ by sampling from

$$P(k_{j,t} = k|\mathbf{t}, \mathbf{k}^{-(j,t)}) \propto m_{*,k}\, \mathcal{L}(k)^{-(j,i)} \tag{8.47}$$

for previously used k, and

$$P(k_{j,t} = K+1|\mathbf{t}, \mathbf{k}^{-(j,t)}) \propto \delta\, \mathcal{L}(k^{new})^{-(j,i)} \tag{8.48}$$

for a newly created menu item $K+1$.

Finally, if the table $t_{j,i}$ is already occupied, we must exclude all counts associated with this table. The samples from the conditional distribution over $k_{j,t}$ are then acquired by sampling from

$$P(k_{j,t} = k|\mathbf{t}, \mathbf{k}^{-(j,t)}) \propto m_{*,k}\, \mathcal{L}(k)^{-(j,i,t)} \tag{8.49}$$

for previously used k, and

$$P(k_{j,t} = K+1|\mathbf{t}, \mathbf{k}^{-(j,)}) \propto \delta\, \mathcal{L}(k^{new})^{-(j,i,t)} \tag{8.50}$$

for a newly created menu item $K+1$, where likelihoods $\mathcal{L}(k)^{-(j,i,t)}$ and $\mathcal{L}(k^{new})^{-(j,i,t)}$ are defined the same way as 8.43 and 8.44, respectively, but excluding counts for all customers sitting at table t.

The last issue that must be clarified is how to create a new frame if a new menu item is selected. It corresponds to sampling from the base distribution χ. We can work on the assumption that the sampling expression for semantic roles is already known (equations 8.24 and 8.26), and design the algorithm 8.1.

The code describes a procedure for creating a new frame f at position u, t. For each slot of frame f, it samples a role from the discrete distribution defined at line 4. The terms describing the probabilities are adopted from equations 8.24 and 8.26 by restricting on the words at current position. If a new role is created, it is necessary to increment the global variable R, which stores the number of active roles.

Algorithm 8.1 Algorithm for sampling a new frame.

1: **procedure** CREATEFRAME(u, t, f) *//Create frame* f *at position* (u, t)
2: **for** $s \in \{1, 2, \ldots, S\}$ **do**
3: $v \leftarrow w_{u,t,s}$
4: Choose role $r_{f,s}$ from the discrete distribution given by

$$r_{f,s} = i \sim \begin{cases} rc_i \dfrac{wc_{v,i}^{-(u,t)} + \beta}{wc_{*,i}^{-(u,t)} + V\beta} & \text{for } 1 \leqslant i \leqslant R \\ \gamma \dfrac{1}{V} & \text{for } i = R+1 \end{cases}$$

5: **if** $r_{f,s} = R+1$ **then**
6: $R \leftarrow R+1$
7: **end if**
8: **end for**
9: **end procedure**

Since the sampling of variables $\mathbf{t}$ and $\mathbf{k}$ separately makes the model more complex than necessary, we will practically use a sampling method, which estimates the values of $f_{j,i} = k_{j,t_{j,i}}$ directly. It can be achieved by grouping together terms associated with sapling k and t

from equations 8.43, 8.44, 8.45, 8.46, 8.47, 8.48 and 8.49 as shown in (Teh et al., 2006):

$$P(f_{u,t} = i|\mathbf{f}^{-(u,t)}, \mathbf{r}, \mathbf{w}, \alpha, \beta, \tau) \propto (fc_{i,u}^{-(u,t)} + \alpha\tau_i) \prod_{s=1}^{S} \frac{wc_{w_{u,t,s},r_{i,s}}^{-(u,t,s)} + \beta}{wc_{*,r_{i,s}}^{-(u,t,s)} + V\beta} \tag{8.51}$$

for $i \in \{1, 2, \ldots, F\}$, and

$$P(f_{u,t} = F + 1|\mathbf{f}^{-(u,t)}, \mathbf{r}, \mathbf{w}, \alpha, \beta, \tau) \propto \alpha\tau_0 \frac{1}{V^S} \tag{8.52}$$

for a new frame. The counts of $m_{*,k}$ and δ are replaced with $\boldsymbol{\tau}$, which is sampled as

$$(\tau_0, \tau_1, \tau_2, \ldots, \tau_F) \sim \mathrm{Dir}(\delta, m_{*,1}, m_{*,2}, \ldots, m_{*,F}). \tag{8.53}$$

The procedure requires to estimate the number of tables sharing the same frame, $m_{*,f}$. It can be obtained by summing the number of tables sharing the same frame in all lexical units:

$$m_{*,f} = \sum_{u=1}^{U} m_{u,f}. \tag{8.54}$$

Antoniak (1974) showed that the number of tables sharing the same frame in a particular lexical unit can be sampled from

$$P(m_{u,f} = l|\mathbf{f}, \mathbf{m}, \tau, \alpha) = \frac{\Gamma(\alpha\tau_f)}{\Gamma(\alpha\tau_f + fc_{f,u})} s(fc_{f,u}, l)(\alpha\tau_0)^l, \tag{8.55}$$

where $s(a, b)$ is the unsigned Stirling number of the first kind, defined as

$$\begin{aligned} s(0,0) &= 1 \\ s(1,1) &= 1 \\ s(a,0) &= 0 \quad \text{for } a > 0 \\ s(a,b) &= 0 \quad \text{for } b > a \\ s(a+1,b) &= s(a,b-1) + a \times s(a,b) \quad \text{else.} \end{aligned} \tag{8.56}$$

8.5 GIBBS SAMPLER FOR NON-PARAMETRIC LDA-FRAMES

At the end of this chapter, we put pieces together and formulate a sampling algorithm for both number of roles and number of frames being unbounded. A simplified sampling procedure is described using the pseudocode in algorithm 8.2.

It begins with the initialization of all random variables and counts at line 1. Line 2 starts the main cycle which repeats the code as

Algorithm 8.2 Algorithm for sampling non-parametric LDA-Frames.

1: InitializeVariables(...)
2: **for** $it \in \{1, 2, \ldots, ITERS\}$ **do**
3: **for** $u \in \{1, 2, \ldots, U\}$ **do** *//Sample frames*
4: **for** $t \in \{1, 2, \ldots T_u\}$ **do**
5: Choose frame $f_{u,t}$ from discrete distribution

$$f_{u,t} = i \sim \begin{cases} (fc_{i,u}^{-(u,t)} + \alpha\tau_i) \prod_{s=1}^{S} \frac{wc_{w_{u,t,s},r_{i,s}}^{-(u,t,s)} + \beta}{wc_{*,r_{i,s}}^{-(u,t,s)} + V\beta} & \text{for } 1 \leqslant i \leqslant F \\ \alpha\tau_0 \frac{1}{V^S} & \text{for } i = F+1 \end{cases}$$

6: **if** $f_{u,t} = F + 1$ **then**
7: $F \leftarrow F + 1$
8: CreateFrame$(u, t, f_{u,t})$
9: **end if**
10: **end for**
11: **end for**
12: **for** $f \in \{1, 2, \ldots, F$ **do** *//Sample roles*
13: **for** $s \in \{1, 2, \ldots S\}$ **do**
14: Choose role $r_{f,s}$ from discrete distribution

$$r_{f,s} = i \sim \begin{cases} rc_{r_{f,s}}^{-(f,s)} \prod_{v=1}^{V} \left(\frac{wc_{v,r_{f,s}}^{-(f,s)} + \beta}{wc_{*,r_{f,s}}^{-(f,s)} + V\beta} \right)^{C_{f,s,v}} & \text{for } 1 \leqslant i \leqslant R \\ \gamma \prod_{v=1}^{V} \frac{1}{V^{C_{f,s,v}}} & \text{for } i = R+1 \end{cases}$$

15: **if** $r_{f,s} = R + 1$ **then**
16: $R \leftarrow R + 1$
17: **end if**
18: **end for**
19: **end for**
20: **for** $u \in \{1, 2, \ldots, U\}$ **do** *//Sample* $m_{u,f}$
21: **for** $f \in \{1, 2, \ldots, F\}$ **do**
22: Choose the number of tables $m_{u,f}$ from

$$P(m_{u,f} = l | \mathbf{f}, \mathbf{m}, \tau, \alpha) = \frac{\Gamma(\alpha\tau_f)}{\Gamma(\alpha\tau_f + fc_{f,u})} s(fc_{f,u}, l)(\alpha\tau_0)^l$$

23: **end for**
24: **end for**
25: Sample $\boldsymbol{\tau}$ from

$$(\tau_0, \tau_1, \tau_2, \ldots, \tau_F) \sim \text{Dir}(\delta, \sum_u m_{u,1}, \sum_u m_{u,2}, \ldots, \sum_u m_{u,F})$$

26: **end for**

many times as many iterations are required. Their number is stored in ITERS.

The code at lines 3 up to 11 is responsible for sampling semantic frames. If a new frame is required to be created, it is generated using the algorithm from 8.1. Semantic roles for each slot of all frames are sampled at lines 12 up to 19. Since the number of frames may change during the inference process, it is necessary to sample new values for variables $m_{u,f}$ at lines 20 up to 24. Consequently, once the new values of $m_{u,f}$ are known, we sample new values of $\boldsymbol{\tau}$ at line 25.

9 HYPERPARAMETER ESTIMATION

After introducing the non-parametric method for an automatic choice of the number of frames and roles in the previous chapter, what remains is to show how to estimate the hyperparameters α, β, γ and δ. The Dirichlet hyperparameters generally have a smoothing effect on the multinomial parameters. Reducing the smoothing effect by lowering the hyperparameters α, β in LDA-Frames model below 1.0 results in making the resulting distributions sparser. It means that the lower the hyperparameter α is, the less frames are used for the description of a particular lexical unit. Analogously, the lower the hyperparameter β is, the less words dominate in a particular semantic role.

The effect of the concentration parameters in Dirichlet processes is similar. Hyperparameters α and δ in the non-parametric model control the prior probability of the number of frames. Hyperparameter γ has an effect on creating new roles. More specifically, the number of tables K in the Chinese restaurant representation of the Dirichlet process with concentration parameter α grows logarithmically with the number of customers n. Antoniak (1974) showed that the expected number of tables that are occupied after n customers have been seated is

$$\mathbb{E}[K|\alpha, n] = \sum_{i=1}^{n} \frac{\alpha}{i-1+\alpha} \approx \alpha \log(n). \tag{9.1}$$

The relationship between hyperparameters and data is symmetric. On one hand, if the model is used to synthesize data, we can control specific properties of the data by tuning the hyperparameters. On the other hand, if the model is supposed to fit unknown data, the hyperparameters should be adjusted to the character of the data.

In the preliminary experiments with LDA-Frames, the hyperparameters have been heuristically set to $\alpha = \beta = 0.1$ for the parametric version, and $\alpha = 5, \beta = 0.1, \gamma = 0.1, \delta = 1$ for the non-parametric version with a good performance. For more elaborate experiments, however, it is convenient to estimate the hyperparameters from data in order to minimize its perplexity. In the subsequent sections, we will show how to estimate α and β in parametric LDA-Frames as well as the hyperparameters in the non-parametric version.

9.1 HYPERPARAMETER ESTIMATION METHOD FOR PARAMETRIC LDA-FRAMES

Although several approaches to learning parameters of Dirichlet priors are known, no exact closed-form solution is available (Ronning, 1989; Wicker et al., 2008). Unfortunately, neither is there a conjugate prior for a straightforward Bayesian inference. Therefore, we use a simple and efficient iterative maximum likelihood approximation derived by Minka (2009).

Dirichlet-Multinomial distribution

The proposed method makes use of the information already available in the collapsed Gibbs sampler, and transforms the problem into the parameter estimation problem for the **Dirichlet-Multinomial distribution**. The Dirichlet-Multinomial distribution is a compound distribution exploiting the conjugacy property of the Dirichlet Distribution and Multinomial distributions, where vector $\mathbf{p}$ is drawn from a Dirichlet distribution with parameter α, and then a sample of discrete outcomes $\mathbf{x}$ is drawn from the multinomial distribution with the probability vector $\mathbf{p}$.

Let us start with the estimation of parameter $\boldsymbol{\alpha}$, which governs the distribution of frames for lexical units. In general, $\boldsymbol{\alpha}$ is an $F-$dimensional vector, where each dimension corresponds to one semantic frame. Because allowing distinct semantic frames to have different α_f parameters may reduce the perplexity of the model, we will describe the general solution first. The solution for the simplified version where $\alpha_1 = \alpha_2 = \cdots = \alpha_F$ is straightforward and will be derived later.

The corresponding Dirichlet-Multinomial distribution $P(\mathbf{f}_u|\alpha)$ is defined as

$$\begin{aligned} P(\mathbf{f}_u|\boldsymbol{\alpha}) &= \int_{\mathbf{p}} P(\mathbf{f}_u|\mathbf{p})P(\mathbf{p}|\boldsymbol{\alpha})\,d\mathbf{p} \\ &= \frac{\Gamma(\sum_{f=1}^{F}\alpha_f)}{\Gamma(\sum_{f=1}^{F}(fc_{f,u}+\alpha_f))}\prod_{f=1}^{F}\frac{\Gamma(fc_{f,u}+\alpha_f)}{\Gamma(\alpha_f)}. \end{aligned} \tag{9.2}$$

The likelihood of the frames across all lexical units can now be expressed as

$$\begin{aligned} P(\mathbf{f}|\alpha) &= \prod_{u=1}^{U} P(\mathbf{f}_u|\boldsymbol{\alpha}) \\ &= \prod_{u=1}^{U}\left(\frac{\Gamma(\sum_{f=1}^{F}\alpha_f)}{\Gamma(\sum_{f=1}^{F}(fc_{f,u}+\alpha_f))}\prod_{f=1}^{F}\frac{\Gamma(fc_{f,u}+\alpha_f)}{\Gamma(\alpha_f)}\right), \end{aligned} \tag{9.3}$$

and the gradient of the log-likelihood as

$$\begin{aligned} g_f &= \frac{d\,\log P(\mathbf{f}|\alpha)}{d\,\alpha_f} = \\ &= \sum_{u=1}^{U}\left[\Psi(\sum_{f=1}^{F}\alpha_f) - \Psi(\sum_{f=1}^{F}fc_{f,u}+\alpha_f) + \Psi(fc_{f,u}+\alpha_f) - \Psi(\alpha_f)\right], \end{aligned}$$

(9.4)

where Ψ is the **digamma function**, defined as the derivative of the gamma function *digamma function*

$$\Psi(x) = \frac{d \log \Gamma(x)}{d\, x}. \tag{9.5}$$

Since there is no analytical solution for expressing the best estimate for α_k, Minka (2009) in his work proposed a fixed-point iteration technique. The idea behind this is to guess an initial value of α_k, and iteratively optimize its value using a lower-bound estimate. This leads to an iteration step expressed as

$$\alpha_f^{new} \leftarrow \frac{\alpha_f \left[\sum_{u=1}^{U} \Psi(fc_{f,u} + \alpha_f) - U\Psi(\alpha_f)\right]}{\sum_{u=1}^{U} \Psi(\sum_{f=1}^{F}(fc_{f,u} + \alpha_f)) - U\Psi(\sum_{f=1}^{F} \alpha_f)}. \tag{9.6}$$

For the LDA-Frames model with symmetric parameter α, we can use the fact that the parameter is just the precision of the Dirichlet distribution φ divided by F (Minka, 2009), and come up with a symmetric version

$$\alpha^{new} \leftarrow \frac{\alpha \left[\sum_{u=1}^{U} \sum_{f=1}^{F} \Psi(fc_{f,u} + \alpha) - UF\Psi(\alpha)\right]}{F \left[\sum_{u=1}^{U} \Psi(\sum_{f=1}^{F}(fc_{f,u} + \alpha)) - U\Psi(F\alpha)\right]}. \tag{9.7}$$

The estimation technique for the hyperparameter β is analogous. We can express the Dirichlet-Multinomial distribution $P(\boldsymbol{w}_r|\boldsymbol{\beta})$ as

$$\begin{aligned} P(\boldsymbol{w}_r|\boldsymbol{\beta}) &= \int_{\mathbf{p}} P(\boldsymbol{w}_r|\mathbf{p})P(\mathbf{p}|\boldsymbol{\beta})\, d\mathbf{p} \\ &= \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\Gamma(\sum_{v=1}^{V}(wc_{v,r} + \beta_v))} \prod_{v=1}^{V} \frac{\Gamma(wc_{v,r} + \beta_v)}{\Gamma(\beta_v)}, \end{aligned} \tag{9.8}$$

the equation for the iterative update of the non-symmetric estimate of β_v as

$$\beta_v^{new} \leftarrow \frac{\beta_v \left[\sum_{r=1}^{R} \Psi(wc_{v,r} + \beta_v) - R\Psi(\beta_v)\right]}{\sum_{r=1}^{R} \Psi(\sum_{v=1}^{V}(wc_{v,r} + \beta_v)) - R\Psi(\sum_{v=1}^{V} \beta_v)}, \tag{9.9}$$

and finally the equation for the estimate of the symmetric version as

$$\beta^{new} \leftarrow \frac{\beta \left[\sum_{r=1}^{R} \sum_{v=1}^{V} \Psi(wc_{v,r} + \beta) - RV\Psi(\beta)\right]}{V \left[\sum_{r=1}^{R} \Psi(\sum_{v=1}^{V}(wc_{v,r} + \beta)) - R\Psi(V\beta)\right]}. \tag{9.10}$$

The hyperparameter estimators can easily be incorporated into the sampling algorithm. We initialize the hyperparameters with a heuristic estimate, e.g. $\alpha = \beta = 0.1$, and update their values at each step of the Gibbs sampler by iterating the assignment from equations 9.6 and 9.9, or from 9.7 and 9.10 until convergence. According to my experience with the model, the convergence is reached after 20 iterations in real applications.

9.2 HYPERPARAMETER ESTIMATION METHOD FOR NON-PARAMETRIC LDA-FRAMES

The non-parametric LDA-frames model is governed by hyperparameters α, β, γ and δ. While the hyperparameter β controls the prior distribution for a finite number of words in semantic roles and can be estimated using the same approach as in case of the parametric model, the other hyperparameters require different methods. We start with the Chinese restaurant process, controlling the number of semantic roles, which is simply being governed by one concentration parameter. The hyperparameters for the hierarchical Dirichlet process controlling the number of frames will be derived later.

The method for estimating γ is an adaptation of the procedure developed by Escobar and West (1995), which places a gamma prior on the values of γ, and incorporates it into the Gibbs sampling analysis. Let us begin with recalling the results of Antoniak (1974), who derived an equation expressing the prior distribution for the number of tables in the Chinese restaurant process

$$P(R|\gamma, n) = c_n(R) n! \gamma^R \frac{\Gamma(\gamma)}{\Gamma(\gamma + n)}, \tag{9.11}$$

where R stands for the number of roles (number of tables in the analogy with the Chinese restaurant), $n = s \times F$, i.e. the number of slots in all frames, corresponding to the number of customers in the Chinese restaurant, and $c_n(R) = P(R|\gamma = 1, n)$, which can be computed using formulas for Stirling numbers if required.

Now, we can omit terms that do not dependent on γ and use the Bayes rule to write

$$P(\gamma|R, n) \propto P(\gamma)P(R|\gamma, n) \propto P(\gamma)\gamma^R \frac{\Gamma(\gamma)}{\Gamma(\gamma + n)}. \tag{9.12}$$

For $\gamma > 0$, the fraction of gamma functions from equation 9.11 can be written as

$$\frac{\Gamma(\gamma)}{\Gamma(\gamma + n)} = \frac{(\gamma + n)\beta(\gamma + 1, n)}{\gamma\Gamma(n)}, \tag{9.13}$$

beta function

where $\beta(a, b)$ is the **beta function** defined as

$$\beta(a, b) = \int_0^1 x^{a-1}(1 - x)^{b-1}\, dx. \tag{9.14}$$

Hence

$$\begin{aligned} P(\gamma|R, n) &\propto P(\gamma)\gamma^{R-1}(\gamma + n)\beta(\gamma + 1, n) \\ &\propto P(\gamma)\gamma^{R-1}(\gamma + n)\int_0^1 x^{\gamma}(1 - x)^{n-1}\, dx. \end{aligned} \tag{9.15}$$

This implies that $P(\gamma|R, n)$ can be viewed as the marginal distribution derived from a joint distribution of γ and a continuous quantity η so that

$$P(\gamma, \eta|R, n) \propto P(\gamma)\gamma^{R-1}(\gamma + n)\eta^{\gamma}(1-\eta)^{n-1}. \tag{9.16}$$

Let γ be drawn from the **gamma distribution** prior with two hyper-prior parameters, shape $a > 0$ and rate $b > 0$, such that *gamma distribution*

$$\gamma \sim \text{Gamma}(a, b) = \frac{b^a x^{a-1} e^{-bx}}{\Gamma(a)}. \tag{9.17}$$

Hence, we have two conditional posterior distributions $P(\eta|\gamma, n)$, $P(\gamma|\eta, n, R, a, b)$ defined as

$$P(\eta|\gamma, n) \propto \eta^{\gamma}(1-\eta)^{n-1} \tag{9.18}$$

and

$$\begin{aligned} P(\gamma|\eta, n, R, a, b) &\propto \gamma^{a-1} e^{-b\gamma} \gamma^{R-1}(\gamma + n)\eta^{\gamma} \\ &\propto \gamma^{a+R-2}(\gamma + n)e^{-\gamma b + \gamma \log(\eta)} \\ &\propto \gamma^{a+R-1} e^{-\gamma(b - \log(\eta))} + n\gamma^{a+R-2} e^{-\gamma(b - \log(\eta))}. \end{aligned} \tag{9.19}$$

Finally, the value of η can be drawn from the beta distribution

$$P(\eta|\gamma, n) \sim \text{Beta}(\gamma + 1, n), \tag{9.20}$$

and the density $P(\gamma|\eta, n, R, a, b)$ reduces to a mixture of two gamma distributions

$$\begin{aligned} P(\gamma|\eta, n, R, a, b) \sim \pi_\eta \text{Gamma}(a + R, b - \log(\eta)) + \\ (1 - \pi_\eta)\text{Gamma}(a + R - 1, b - \log(\eta)) \end{aligned} \tag{9.21}$$

with weight π_η defined as

$$\begin{aligned} \frac{\pi_\eta}{1 - \pi_\eta} &= \frac{a + R - 1}{n(b - \log(\eta))} \Rightarrow \\ \pi_\eta &= \frac{a + R - 1}{a + R - 1 + n(b - \log(\eta))}. \end{aligned} \tag{9.22}$$

Since the direct sampling from a weighted sum of two distributions is not convenient, we will first sample

$$c \sim \text{Bern}(\pi_\eta), \tag{9.23}$$

and then sample γ based on the outcome of c:

$$P(\gamma|\eta, n, R, a, b) \sim \begin{cases} \text{Gamma}(a + R, b - \log(\eta)) & \text{if } c = 1 \\ \text{Gamma}(a + R - 1, b - \log(\eta)) & \text{else.} \end{cases} \tag{9.24}$$

The complete sampling scheme for the hyperparameter γ is described in algorithm 9.1. The algorithm performs successive Gibbs samples

Algorithm 9.1 Algorithm for sampling the γ hyperparameter.

```
procedure SampleGamma(n, R, a, b, ITERS)
    SUM ← 0
    for i ∈ {1, 2, ..., ITERS} do
        Sample η ~ Beta(γ + 1, n)
        π_η ← (a + R − 1) / (a + R − 1 + n(b − log(η)))
        Choose c ← Bern(π_η)
        if c=1 then
            SUM ← SUM + Gamma(a + R, b − log(η))
        else
            SUM ← SUM + Gamma(a + R − 1, b − log(η))
        end if
    end for
    γ ← SUM / ITERS
end procedure
```

and averages their outcomes. It requires two hyper-prior parameters. According to the experiments a good choice is $a = b = 1$, however their values have little effect on the resulting number of roles.

The same approach as in case of γ hyperparameter can be used for sampling the δ hyperparameter, which serves as the concentration parameter of a single Dirichlet process as well. The sampling scheme for the α hyperparameter governing the second-level Dirichlet process, however, requires a slight modification. We will follow the method proposed by Teh et al. (2006) that is a straightforward extension of the single Dirichlet process procedure.

Again, let J be the number of Chinese restaurants with $n_{j,*}$ customers and $m_{j,*}$ tables in the jth restaurant. The concentration parameter α governs the number of tables in the restaurants in the following way:

$$P(m_{1,*}, m_{2,*}, \ldots, m_{J,*} | \alpha, n_{1,*}, n_{2,*}, \ldots, , n_{J,*}) = \prod_{j=1}^{J} s(n_{j,*}, m_{j,*}) \alpha^{m_{j,*}} \frac{\Gamma(\alpha)}{\Gamma(\alpha + n_{j,*})}. \tag{9.25}$$

We assume that the prior for α is again the gamma distribution with hyper-prior parameters a, b. Moreover, we define auxiliary variables $\boldsymbol{w} = (w_1, w_2, \ldots, w_J)$ and $\mathbf{s} = (s_1, s_2, \ldots, s_J)$. For each j we can write

$$\frac{\Gamma(\alpha)}{\Gamma(\alpha + n_{j,*})} = \frac{1}{\Gamma(n_{j,*})} \int_0^1 w_j^{\alpha} (1 - w_j)^{n_{j,*} - 1} \left(1 + \frac{n_{j,*}}{\alpha}\right) dw_j, \tag{9.26}$$

and hence the joint distribution of $\alpha, \boldsymbol{w}$ and $\mathbf{s}$ is

$$P(\alpha, \boldsymbol{w}, \mathbf{s}|\mathbf{m}, \mathbf{n}) \propto$$
$$\alpha^{a-1+m_{*,*}} e^{-\alpha b} \prod_{j=1}^{J} w_j^{\alpha}(1-w_j)^{n_{j,*}-1} \left(\frac{n_{j,*}}{\alpha}\right)^{s_j}. \tag{9.27}$$

We can now normalize the joint distribution to get the conditional distribution $P(\alpha|\boldsymbol{w}, \mathbf{s}, \mathbf{m}, \mathbf{n})$

$$P(\alpha|\boldsymbol{w}, \mathbf{s}, \mathbf{m}, \mathbf{n}) \propto \alpha^{a-1+m_{*,*}-\sum_{j=1}^{J} s_j} e^{-\alpha(b-\sum_{j=1}^{J} \log(w_j))}, \tag{9.28}$$

which can be expressed as a gamma distribution

$$P(\alpha|\boldsymbol{w}, \mathbf{s}, \mathbf{m}, \mathbf{n}) \propto \text{Gamma}(a + m_{*,*} - \sum_{j=1}^{J} s_j, b - \sum_{j=1}^{J} \log(w_j)). \tag{9.29}$$

Given α, the $\boldsymbol{w}$ and $\mathbf{s}$ variables are conditionally independent, and we can write

$$P(w_j|\alpha, \mathbf{n}) \propto w_j^{\alpha}(1-w_j)^{n_{j,*}-1} \propto \text{Beta}(\alpha+1, n_{j,*}), \tag{9.30}$$

and

$$P(s_j|\alpha, \mathbf{n}) \propto \left(\frac{n_{j,*}}{\alpha}\right) \propto \text{Bern}(\frac{n_{j,*}}{\alpha}). \tag{9.31}$$

The complete sampling procedure for hyperparameter α is described in algorithm 9.2.

Algorithm 9.2 Algorithm for sampling the α hyperparameter.

```
procedure SampleAlpha(U, T_1, ..., T_U, m_{*,*}, a, b, ITERS)
    SUM ← 0
    for i ∈ {1, 2, ..., ITERS} do
        w ← 0
        s ← 0
        for u ∈ {1, 2, ..., U} do
            w ← w + Beta(α + 1, T_u)
            s ← s + log(Bern(T_u / α))
        end for
        SUM ← SUM + Gamma(a + m_{*,*} − s, b − w)
    end for
    α ← SUM / ITERS
end procedure
```

10 DISTRIBUTED INFERENCE ALGORITHMS

With the increasing availability of big data, we need efficient methods to process it. Many promising and sophisticated algorithms in artificial intelligence and natural language processing are beaten by simple algorithms just because the simple algorithms are usually very efficient and can be run on much larger data sets in reasonable time. Problems that seemed to be solved years ago reappear as new challenges when the amount of input data rises by orders of magnitude.

Fortunately, as the amount of the data grows, the availability of computational resources grows as well. The solution, however, is not straightforward. While the overall amount of Central Processing Unit (CPU) power for a unit price increases, the maximum clock rate of a single CPU based on current technologies reaches its limits coming from the physical laws. This opens up a problem of redistributing the computation among several processors, and then merging the results into one solution.

This chapter is devoted to studying the problem of distributing the inference algorithm for LDA-Frames and non-parametric LDA-Frames among a multitude of processors on a single computer. Unlike the distribution in a cluster of different computers, a parallel algorithm on a single computer can make use of a shared memory quickly accessible by any CPU. This makes the implementation easier, nevertheless, the described principles are applicable on clusters of computers as well.

10.1 PARALLEL LDA-FRAMES

The easiest and the most straightforward way to distribute the inference process is to parallelize the non-collapsed Gibbs sampler. Since the samples of $f_{u,t}$ given $\mathbf{r}, \boldsymbol{w}, \boldsymbol{\varphi}, \boldsymbol{\theta}$, as well as the samples of $r_{f,s}$ given $\mathbf{f}, \boldsymbol{w}, \boldsymbol{\theta}$ are conditionally independent, they can be sampled concurrently. Specifically, we would concurrently sample

$$\begin{aligned} &f_{u,t} \mid \mathbf{r}, \boldsymbol{w}, \boldsymbol{\varphi}, \boldsymbol{\theta} \\ &r_{f,s} \mid \mathbf{f}, \boldsymbol{w}, \boldsymbol{\theta} \\ &\boldsymbol{\varphi}_u \mid \mathbf{f} \\ &\boldsymbol{\theta}_r \mid \mathbf{r}, \boldsymbol{w}, \mathbf{f}. \end{aligned} \tag{10.1}$$

The problem, however, is that the non-collapsed samplers converge very slowly. In general, the more variables are collapsed the faster convergence of the inference process can be expected (Liu et al.,

1994; Newman et al., 2009). The slow convergence motivates us to distribute the fully collapsed sampler, which takes the advantage of efficiency. A difficulty of distributing the fully collapsed sampler, however, lies in the fact that neither the samples of frames nor the samples of roles are conditionally independent. This results in an approximate solution. Fortunately, as will be shown in chapter 11, under some conditions, the approximation has little effect on the quality of the resulting model.

We begin with dividing the inference problem into two parts – the inference of latent variables $f_{u,t}$ and the inference of variables $r_{f,s}$. The simplest partitioning of the $\mathbf{f}$ variables is to randomly distribute the U lexical units over P processors with approximately $\frac{U}{P}$ lexical units per processor such that

$$\begin{aligned}\mathbf{f} &= \{f_{1,1} \dots f_{l,T_l}\}_1 \cup \{f_{l+1,1} \dots f_{m,T_m}\}_2 \cup \dots \cup \{f_{n,1} \dots f_{U,T_U}\}_P \\ &= \mathbf{pu}_1 \cup \mathbf{pu}_2 \cup \dots \cup \mathbf{pu}_P\end{aligned} \tag{10.2}$$

Analogously, the set of latent variables $\mathbf{r}$ is distributed based on the frame identifiers, i.e.

$$\begin{aligned}\mathbf{r} &= \{r_{1,1} \dots r_{l,S}\}_1 \cup \{r_{l+1,1} \dots r_{m,S}\}_2 \cup \dots \cup \{r_{n,1} \dots r_{F,S}\}_P \\ &= \mathbf{pf}_1 \cup \mathbf{pf}_2 \cup \dots \cup \mathbf{pf}_P.\end{aligned} \tag{10.3}$$

We can now distinguish two approaches to parallelized sampling:

- **synchronous**, where the processors have independent memory spaces and the synchronization is done at the end of each iteration;
- **asynchronous**, where the processors share the memory and alter the counts simultaneously.

In order to be able to describe the synchronous approach, let us introduce processor-specific count variables $wc_{v,x}(p)$ and $wc_{*,x}(p)$ which are defined the same way as 7.21 and 7.22, respectively, but store only a local copy of data for processor p. The parallelized procedure is described in algorithm 10.1.

The algorithm starts with sampling all frames in a distributed manner, synchronizes all counts, and continues with sampling roles. The cycles at lines 3 up to 10 and 12 up to 18 represent a parallelized execution of the code. Notice that at the beginning of each parallel section, the global counts $\boldsymbol{wc}$ are copied to the local counts $\boldsymbol{wc}(p)$, which remain private until the end of the parallel blocks. As soon as all blocks are finished, the local counts are synchronized into the global counts $\boldsymbol{wc}$.

Algorithm 10.1 Algorithm for distributed LDA-Frames.

1: INITIALIZEVARIABLES(...)
2: **for** $it \in \{1, 2, \ldots, ITERS\}$ **do**
3: **for** $p \in P$ **do** *in parallel*
4: Copy global counts $\boldsymbol{wc}(p) \leftarrow \boldsymbol{wc}$
5: **for** $u \in \mathbf{pu}_p$ **do** *//Sample frames*
6: **for** $t \in \{1, 2, \ldots T_u\}$ **do**
7: Choose frame $f_{u,t}$ from discrete distribution

$$f_{u,t} = i \sim (fc_{i,u}^{-(u,t)} + \alpha) \prod_{s=1}^{S} \frac{wc_{w_{u,t,s}, r_{i,s}}^{-(u,t,s)}(p) + \beta}{wc_{*, r_{i,s}}^{-(u,t,s)}(p) + V\beta}$$

8: **end for**
9: **end for**
10: **end for**
11: Synchronize global counts $\boldsymbol{wc} \leftarrow \boldsymbol{wc} + \sum_{p=1}^{P} (\boldsymbol{wc}(p) - \boldsymbol{wc})$
12: **for** $p \in P$ **do** *in parallel*
13: Copy global counts $\boldsymbol{wc}(p) \leftarrow \boldsymbol{wc}$
14: **for** $f \in \mathbf{pf}_p$ **do** *//Sample roles*
15: **for** $s \in \{1, 2, \ldots S\}$ **do**
16: Choose role $r_{f,s}$ from discrete distribution

$$r_{f,s} = i \sim \prod_{v=1}^{V} \left(\frac{wc_{v, r_{f,s}}^{-(f,s)}(p) + \beta}{wc_{*, r_{f,s}}^{-(f,s)}(p) + V\beta} \right)^{C_{f,s,v}}$$

17: **end for**
18: **end for**
19: **end for**
20: Synchronize global counts $\boldsymbol{wc} \leftarrow \boldsymbol{wc} + \sum_{p=1}^{P} (\boldsymbol{wc}(p) - \boldsymbol{wc})$
21: **end for**

Because the Gibbs sampling is strictly a sequential process, in order to properly sample from the posterior distribution, the updates should not be performed concurrently with the update of the counts after each iteration. However, because there is typically a large number of lexical units and semantic roles in comparison to the number of processors, the dependence between latent variables $\mathbf{f}$ and $\mathbf{r}$ is weak and the approximation is almost equal to the exact algorithm.

Notice that the synchronous algorithm can also be simply used for distributed computation on a cluster of different computers. It only requires synchronization of the counts between all nodes after each iteration.

Asynchronous distributed computation of LDA-Frames has an even simpler implementation than the synchronous algorithm. It uses a shared memory for storing all counts, and updates them simultaneously. It is only required to be properly treated with mutual exclusions. Despite the fact that the mutual exclusions can block the execution of the code and cause a longer running time of the algorithm, the immediate update of counts results in a better approximation and is preferred. Unfortunately, the asynchronous approach is hardly applicable in the distributed computation on a cluster of different computers because the synchronization after every change of a count variable between all computers is too expensive.

10.2 PARALLEL NON-PARAMETRIC LDA-FRAMES

The design of a parallel algorithm for the non-parametric LDA-Frames is quite complicated. Unlike the parametric case which uses a fixed set of frames and roles, individual processors in the non-parametric case may instantiate both new frames and new roles during each iteration. This leads to several complications that need to be settled:

- **merging the roles** – because letting the newly created roles from each processor be distinct would lead to unnecessary growth in the total number of roles, it is useful to merge them during the synchronization phase to produce a more compact model. The question, however, is how to link the roles from different processors. A simple way is to merge new roles based on their integer identifier. For example, suppose that three processors produced four, three and six new roles, respectively. Then during the synchronization, the roles would be aligned by their topic label, which would result in six global roles. Although this solution is computationally simple and efficient, it is seems to be very suboptimal. To improve the merging process, Newman et al. (2009) in his work on distributed algorithms for topic models, proposed aligning the topics of the Hierarchical Dirichlet process using a similarity function such as the Kullback-Leibner divergence. Finding the optimal match, though, is NP-hard for cases where the number of processors is greater than two (Burkard and Çela, 1999).

- **merging the frames** – sampling frames on multiple processors faces the same problem as in the case of roles. The alignment, however, should not be based on frame identifiers. Rather, it requires exact matching of their semantic roles. When two processors produce identical frames, it is necessary to merge them in order to avoid global identical frames with distinct frame identifiers.

- **asynchronous algorithm** – while the asynchronous parallel algorithm for the parametric LDA-Frames is very straightforward and efficient, the design of an efficient asynchronous algorithm for the non-parametric LDA-Frames is almost impossible. Because the shared set of frames and roles need to be iterated as well as updated on each processor simultaneously, the synchronization and the mutual exclusion mechanism would be so complex that the benefit of the parallelization would be very small.

Since the parallelization of the non-parametric LDA-Frames brings many complications that require to design a new algorithm from scratch, its practical implementation is left for the further work.

11 EVALUATION

A key part of the development of all models regardless their application is an evaluation. The evaluation is a systematic determination of the model's merit using a set of criteria. A natural way to evaluate probabilistic models is either estimating the probability of unseen held-out data given a trained model, or measuring the performance on some secondary task. In this chapter, we will first focus on evaluating the models using perplexity (which is strongly related to probability), and using their reconstruction abilities. Several applications of LDA-Frames along with their evaluation will be discussed in the subsequent sections.

11.1 DATA PREPARATION

Although the LDA-Frames method for semantic frame induction is language independent in general, we will mainly focus on English, which belongs among the most studied world's languages. It enables us to compare the LDA-frames against other English lexical resources.

Although there are many large web-based text corpora of English, currently available on-line for free[1], we will start with a much smaller, but very well studied and described corpus known as the British National Corpus (BNC), XML edition released in 2007. It is a 100 million word collection of texts of both written and spoken British English, mainly from the end of the 20th century. The size is large enough for training meaningful semantic frames, and small enough, at the same time, for efficient and quick experiments. An important property of the corpus is its thematic balance, covering many different styles and topics.

The most challenging part of the training data acquisition process is the syntactic parsing phase. Particularly for English, we use the dependency parses provided by the Stanford Parser (Klein and Manning, 2003).

The parser is written in the Java programming language, and implements a lexicalized Probabilistic Context-Free Grammar (PCFG) parser. The dependency parses are automatically extracted from phrase structure parses with reported accuracy 80.3 % using the method described in de Marneffe et al. (2006).

1 For example the Common Crawl corpus, released by the Common Crawl Foundation, http://commoncrawl.org/.

While the phrase structure parse represents nesting of multi word constituents, the dependency parse represents dependencies between individual words. The typed dependency parsing, which is required for acquiring the training data, additionally labels dependencies with grammatical relations, such as `subject`, `direct object`, `indirect object`, `modifier`, etc. An example of the dependency parse for the sentence "I saw the man who loves you." provided by the system is shown in figure 11.1.

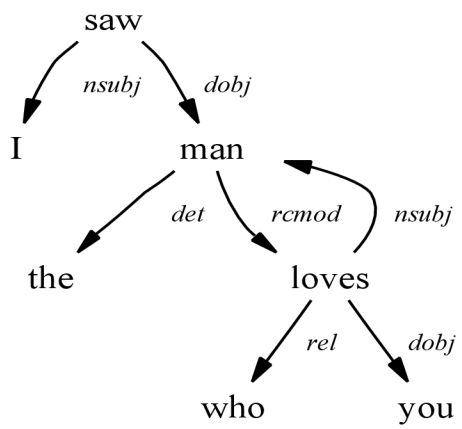


Figure 11.1: Example of the typed dependency parse for the sentence "I saw the man who loves you".

governor
head
dependent

The current Stanford parser supports approximately 50 grammatical relations (the actual number depends on some options). The dependencies are binary relations between a **governor**, also known as **head**, and a **dependent**. The complete list of supported grammatical relations can be found in the Stanford dependency manual (de Marneffe and Manning, 2008). The most important relations useful for generating the training data are defined and exemplified below in alphabetical order. Bolded words in example sentences with subscripts 'H' and 'D' denote the head and the dependent, respectively:

- ***advmod:* adverbial modifier**
  An adverbial modifier of a word is a (non-clausal) adverb or adverbial phrase (ADVP) that serves to modify the meaning of the word.

(11.1) ***Genetically***$_{\mathrm{D}}$ ***modified***$_{\mathrm{H}}$ *food.*

- ***agent:* agent**
  An agent is the complement of a passive verb which is introduced by the preposition 'by' and does the action. It does not appear in basic dependencies output.

(11.2) *The man has been* ***killed***$_{\mathrm{H}}$ *by the* ***police***$_{\mathrm{D}}$.

- ***amod:*** **adjectival modifier**
  An adjectival modifier of an NP is any adjectival phrase that serves to modify the meaning of the NP.

(11.3) *Sam eats* ***red***$_{\mathrm{D}}$ ***meat***$_{\mathrm{H}}$.

- ***dobj:*** **direct object**
  A direct object of a VP is the noun phrase which is the (accusative) object of the verb.

(11.4) *They* ***win***$_{\mathrm{H}}$ *the* ***lottery***$_{\mathrm{D}}$.

- ***iobj:*** **indirect object**
  An indirect object of a VP is the noun phrase which is the (dative) object of the verb.

(11.5) *She* ***gave***$_{\mathrm{H}}$ ***me***$_{\mathrm{D}}$ *a raise.*

- ***nsubj:*** **nominal subject**
  A nominal subject is a noun phrase which is the syntactic subject of a clause. The governor of this relation might not always be a verb: when the verb is a copular verb, the root of the clause is the complement of the copular verb, which can be an adjective or noun.

(11.6) ***Clinton***$_{\mathrm{D}}$ ***defeated***$_{\mathrm{H}}$ *Dole.*

- ***nsubjpass:*** **passive nominal subject**
  A passive nominal subject is a noun phrase which is the syntactic subject of a passive clause.

(11.7) ***Dole***$_{\mathrm{D}}$ *was* ***defeated***$_{\mathrm{H}}$ *by Clinton.*

Although the LDA-Frames algorithm is designed to work with predicates of any parts of speech, we will mainly focus on verbs, which are considered to be the most important constituents of sentences in natural languages, and are described in many valency lexicons that can be used for the comparison purposes. For the testing and evaluation of the algorithm, we will use three data sets, named *BNC-big*, *BNC-small*, and *synthetic*.

To create the *BNC-big* data set, we extracted all occurrences of transitive verbs from BNC, where both the *nsubj* and *dobj* relations were presented. This resulted in a list of triples, where the first component of the triple represents a verb predicate, the second component represents a realization of subject, and the last component represents a direct object. The positions where the subject, direct object or both were missing were skipped. The list of transitive verbs consists of 4576 predicates, and was obtained from the Corpus Pattern Analysis project (Hanks and Pustejovsky, 2005). In order to

remove the noise and misspellings, we further filtered out all tuples, where the subject or direct object realization occurred fewer than three times in the corpus. This procedure resulted in a list of 1,447,984 data tuples.

Because some of the further experiments require a fast computation, we have created a subset of *BNC-big* called *BNC-small*, which consists of 240 predicates in 285,598 data tuples. The tuples have been selected so that the predicate frequency had had to be greater than 1000 and smaller than 8000, and the frequency of role realizations had had to be greater than 500. The upper limit for the predicate frequency is used to filter out too general and frequent verbs that usually do not evoke any remarkable frames.

Both *BNC-big* and *BNC-small* data sets are further split into a training part and testing part, where the training part accounts for 90 % of the tuples and the test part 10 % of the tuples.

The data set called *synthetic* is created artificially from a known set of semantic frames and roles in order for us to be able to evaluate whether the algorithm is able to reconstruct them. The core of the data is built from a set of four semantic roles, each of them with several lexical realizations. The set of roles is given by table 11.1. Notice, that some of the realizations (for instance 'fish' or 'jenny') are assigned to more than one semantic role. This makes the semantic roles more realistic and the realization–role assignment ambiguous.

semantic roles	**realizations**
Animal	dog, cat, mouse, fish, chicken, jenny, rabbit
Food	cake, fish, lunch, chicken, dinner, meat, bread, rabbit
Institution	school, state, university, police, company
Person	people, man, woman, john, jenny

Table 11.1: Semantic roles along with their lexical realizations.

The set of lexical units consists of eleven verbs, each of them connected with several semantic frames corresponding to the `subject` and `object` grammatical relations. These frames are only composed of the roles defined in table 11.1, and therefore have no ambition to be complete (in the sense that there are no more frames for a lexical unit). The selected lexical units with the corresponding frames are listed in table 11.2. It is worth mentioning, that two of the frames have the sign '-' instead of a semantic role. This stands for an empty slot without an obligatory semantic role at the corresponding

grammatical relation position. Such slots are excluded from the inference process.

Lexical unit	**Semantic frames**
eat	(Person, Food), (Animal, Food)
cook	(Person, Food)
chew	(Animal, Food), (Person, Food)
bite	(Animal, Food), (Animal, -)
paint	(Person, Person), (Person, -)
teach	(Person, Person), (Institution, Person), (Person, Animal)
love	(Person, Person)
hire	(Institution, Person)
pay	(Institution, Person)
produce	(Institution, Food)
buy	(Institution, Food), (Person, Food), (Person, Animal)
feed	(Person, Animal), (Person, Person)
train	(Person, Animal)
look_after	(Person, Animal), (Person, Person)
work_for	(Person, Institution)
sue	(Person, Institution), (Institution, Institution)

Table 11.2: Lexical units and their semantic frames.

The data set of frame realization tuples have been generated using the semantic roles and frames from tables 11.1, 11.2 according to the following procedure.

For every lexical unit u:

- Choose a number of corpus realizations $N_u \in \{5, \ldots, 25\}$ from the uniform distribution.
- For each realization $n_u \in \{1, \ldots, N_u\}$, among all permitted frames for lexical unit u, choose a semantic frame f_{n_u} from the uniform distribution.
- For each frame f_{n_u} generate realizations for all its roles according to table 11.1 using the uniform distribution.

The procedure produced a data set consisting of 160 subject-object realization tuples. An important quality of this data set is that we know which frame is responsible for generating a particular realization. However, this information is not presented in the generated data.

To conclude, the data sets used for the evaluation purposes in this chapter are summarized in table 11.3. The second column

data set	size	lexical units	vocabulary size
BNC-big	1,447,984	4,053	28,990
BNC-big-train	1,303,186	4,053	28,911
BNC-big-test	144,798	4,053	18,870
BNC-small	285,599	240	867
BNC-small-train	257,040	240	867
BNC-small-test	28,559	240	867
synthetic	160	16	14

Table 11.3: Evaluation data sets.

named "size" represents the number of all tuples in the data, column "lexical units" represents the number of unique predicates, and the last column "vocabulary size" summarizes the number of unique semantic role realizations regardless the slot a realization is associated with.

11.2 IMPLEMENTATION

The implementation of LDA-Frames and the related models can be divided into two parts. One part is responsible for the data preparation and is implemented in Python, the second part implements the sampling algorithms described in chapters 7, 8, 9, 10 and is written in the C++ programming language. For the syntactic parsing of BNC a third-party implementation of the Stanford parser written in Java[2] has been used.

The directory tree structure of the code as is provided on the attached CD as well as in the Google code repository[3] is depicted in figure 11.2. The directory `3rdparty` contains binaries of the Stanford CoreNLP package, version 1.3.4[4], implementing the Stanford parser along with other important tools for processing English texts. The third-party binaries are not provided in the Google code repository. The data for evaluation is provided in directory `data`. Scripts for manipulating and accessing the generated semantic frames are stored in `libldaf`. All tools intended for preprocessing data are included in `preprocessing`. It also contains a wrapper for the Stanford parser written in Python. The core of the project is located in the `sampler` folder. It implements the LDA-Frames and Non-parametric samplers

2 http://nlp.stanford.edu/software/lex-parser.shtml
3 http://code.google.com/p/lda-frames/
4 http://nlp.stanford.edu/software/corenlp.shtml

```
root
├── 3rdparty
│   └── stanford-corenlp
├── data
│   ├── BNC-big
│   ├── BNC-small
│   └── synthetic
├── libldaf
├── preprocessing
│   └── stanford-parser
├── sampler
└── utils
```

Figure 11.2: Directory tree structure.

written in C++. Finally, a script for the evaluation of frames created from the synthetic data, and a script for generating random data are placed into the `utils` directory.

The input data for the sampler must have the form of a text file where each line corresponds to one lexical unit. Each line consists of tabulator-separated corpus realizations, where each realization is composed of space-separated positive integers that represent word identifiers for all slots. The number 0 is reserved for an empty slot. An example of the input file for 3 lexical units with two-slot realizations is shown below.

```
1 2     3 4     2 3
4 2     1 3     3 2     3 2
4 4     3 4     4 0
```

In order to be able to interpret the results, a dictionary that translates line numbers to lexical units and word identifiers to words is required. The sampler input file along with the dictionary can be generated using the `generate_samplerinput.py` script stored in the `preprocessing` folder. As its input, it takes a text file where each line consists of a lexical unit and tabulator-separated realizations from the corpus. The first line must start with ';' and must consist of tabulator-separated names of used grammatical relations. For example:

```
;SUBJECT    OBJECT
eat         people          cake
eat         man             bread
cook        woman           cake
```

The generated result produced by the sampler comprises two files: `frames.smpl`, which contains frame identifiers corresponding to the realization positions in the input file, and `roles.smpl`, which contains

as many rows as the number of frames, and as many columns as the number of slots. The numbers there represent semantic role identifiers.

Instructions for the source codes compilation and examples of the usage are provided in appendix A. A demo application of the LDA-Frames data inferred from the *BNC-big* set is operated on the LDA-Frames home page[5]. The data has been computed for 1200 semantic frames and 400 semantic roles. Semantic frames, sorted by their probability scores in descending order, are represented by their semantic role numbers and their top ten realizations. A screenshot of an example for lexical unit "eat" taken on the LDA-Frames web site is shown in figure 11.3.

EAT

	SUBJECT		OBJECT	
	222		40	
0.554086 frame 1166	0.794216	person	0.085888	food
	0.010335	people	0.046396	meal
	0.007963	one	0.01947	egg
	0.005797	man	0.01947	breakfast
	0.004342	who	0.01726	lunch
	0.003409	woman	0.016846	dinner
	0.002687	child	0.015189	fish
	0.002519	that	0.013256	meat
	0.002307	all	0.012289	potato
	0.002215	someone	0.012151	cake
	152		40	
0.128011 frame 622	0.027104	bird	0.085888	food
	0.026926	dog	0.046396	meal
	0.023538	animal	0.01947	egg
	0.023181	fish	0.01947	breakfast
	0.016049	cat	0.01726	lunch
	0.014979	child	0.016846	dinner
	0.013374	people	0.015189	fish
	0.01266	prey	0.013256	meat
	0.011947	man	0.012289	potato
	0.011769	horse	0.012151	cake

Figure 11.3: A screenshot of the semantic frame representation on the LDA-Frames web site – two most likely frames for lexical unit 'eat'.

5 http://nlp.fi.muni.cz/projekty/lda-frames

11.3 PERPLEXITY

The term **perplexity** came from the information theory and refers to the measurement of how well a probability model predicts a sample of data. The sample is usually taken from unseen data sets, and thus the perplexity can be used to compare probability models. In the context of LDA-Frames models, the perplexity is used to evaluate the convergence as well as the estimation of parameters. *perplexity*

Let $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$ denote a test data sample and m a probabilistic model. The perplexity of data $\mathbf{x}$ given model m is defined as

$$\text{Perplexity}(\mathbf{x}, m) = 2^{-\frac{1}{N}\sum_{i=1}^{N} \log_2(P(x_i|m))} . \tag{11.1}$$

As one can see from the equation, better models assign a higher probability to the test data, which results in lower perplexity. Moreover, the exponent of the equation can be regarded as the average number of bits required to represent a test event x_i in an optimal code based on m. If the model m was a uniform distribution over k discrete events, the perplexity would be k.

In order to be able to compute the perplexity of the LDA-Frames model, we need to express the probability of a word from a test set given a model inferred on a train model. Let $\boldsymbol{w}'$ denote words from the test set. The perplexity can be derived as follows:

$$\text{Perplexity}(\boldsymbol{w}', m) = 2^{-\frac{1}{|\boldsymbol{w}'|}\sum_{u=1}^{U}\sum_{t=1}^{T_u}\sum_{s=1}^{S} \log_2 P(w'_{u,t,s})} \tag{11.2}$$

$$\begin{aligned} P(w'_{u,t,s}) &= \sum_{f=1}^{F} P(f|u)P(w'_{u,t,s}|f) = \\ &= \sum_{f=1}^{F} P(f|u)P(w'_{u,t,s}|\theta_{r_{f,s}}) = \\ &= \sum_{f=1}^{F} \left[\frac{fc_{f,u} + \alpha_f}{\sum_{f=1}^{F}(fc_{f,u} + \alpha_f)} \times \frac{wc_{w'_{u,t,s}, r_{f,s}} + \beta_{w'_{u,t,s}}}{\sum_{v=1}^{V}(wc_{w'_{u,t,s}, r_{f,s}} + \beta_{w'_{u,t,s}})} \right], \end{aligned} \tag{11.3}$$

where m stands for the LDA-Frames model computed using a training set $\boldsymbol{w}$.

11.3.1 *Convergence*

In the first type of the evaluation, we will check the convergence properties of LDA-Frames. The model is always trained on the *train*

subset only using the single-core algorithm with hyperparameter estimation. The perplexity is measured on both *train* and *test* subsets in order to be able to reveal overfitting.

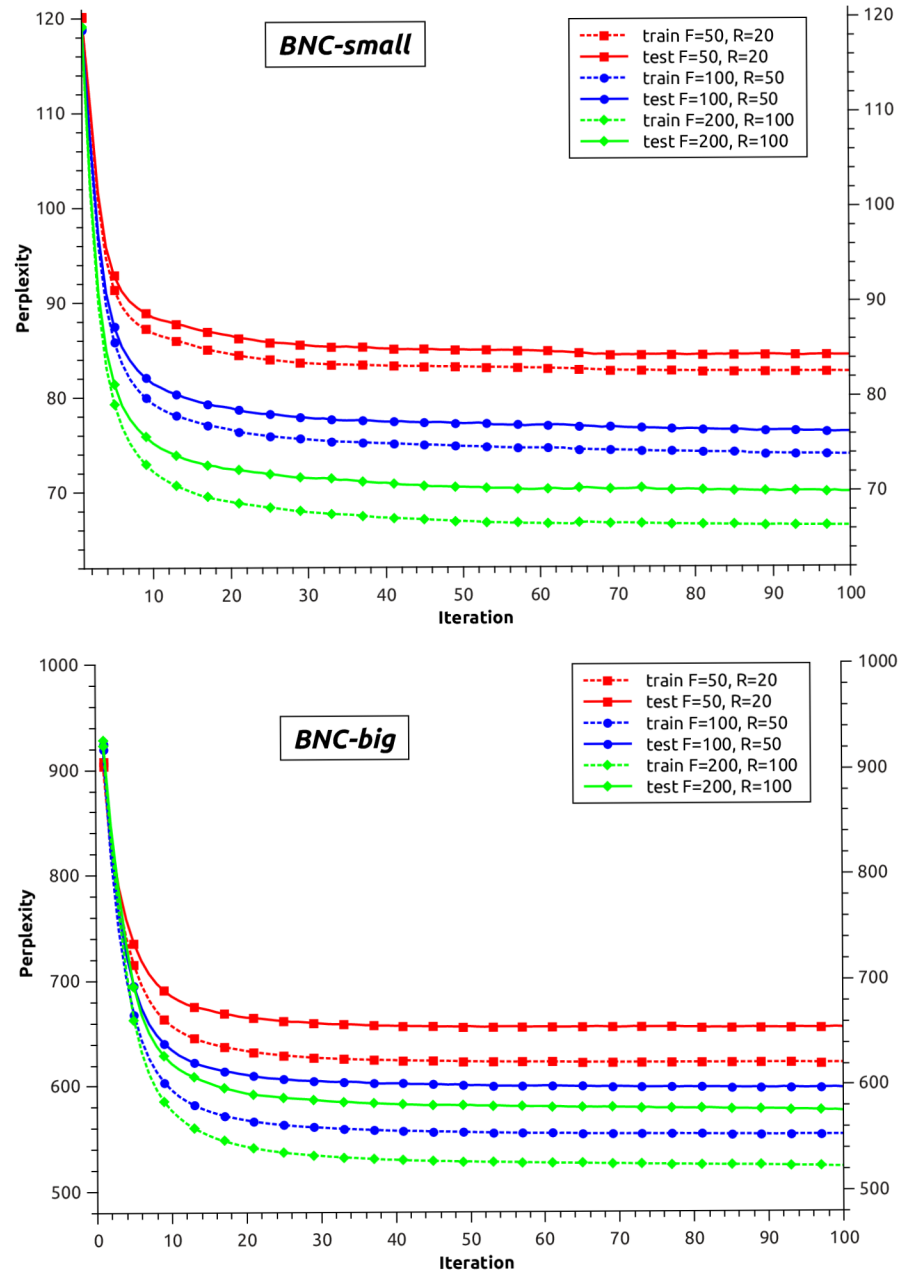


Figure 11.4: Convergence of the LDA-Frames algorithm.

The convergence of the parametric LDA-Frames algorithm is illustrated in Figure 11.4. It has been tested on both *BNC-small* and *BNC-big* data sets for three different settings. As one can see from the graphs, the perplexity steeply decreases during the first 50 iterations and then continues to decrease very slowly. After approximately 100 iterations, it reaches its minimum. Although the perplexity of the

training subset is always lower than the case of the test subset, the perplexity of the test set is monotonously decreasing during the whole learning process. It confirms that the model is not predisposed to overfitting.

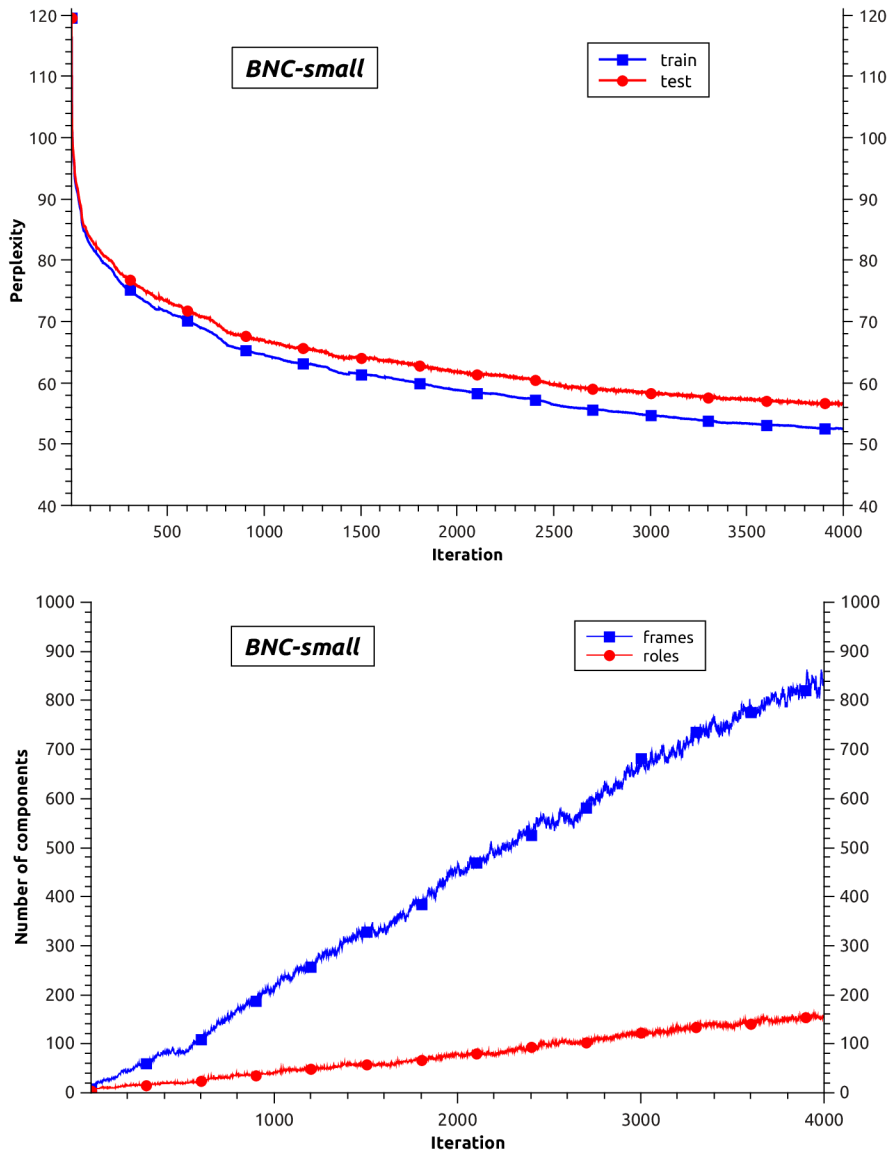


Figure 11.5: Convergence and the number of components in the Non-Parametric LDA-Frames algorithm.

The convergence of the non-parametric LDA-Frames with hyper-parameter estimation is depicted in the top graph in figure 11.5. The figure shows that the convergence is much slower than the case of parametric LDA-frames, but it continues to decrease deeply below the lowest value of the parametric cases. Both the slow convergence and better values of perplexity are caused by the reestimation of the number of frames and roles during the inference process, which starts

at $F = R = 1$. The bottom graph in figure 11.5 shows how the number of frames and roles evolves during the inference process. One can see again that the model is not overfitted.

11.4 RECONSTRUCTION ABILITIES

An important question, however, is whether the estimated values of the number of frames and roles are optimal. In order to test the hypothesis, an experiment with the *synthetic* data set has been carried out. Since the data is created using a procedure with known frames and roles, we can compare the inferred values against the original ones. First, the parametric LDA-Frames algorithm with hyperparameter estimation and the number of frames and roles parameters set to $R \in \{1 \ldots 10\}$, $F \in \{1 \ldots 20\}$ has been run on the *synthetic* data set. This experiment produced 200 different models. Their perplexities reached after 1000 iterations of the inference algorithm are depicted in the graph in figure 11.6.

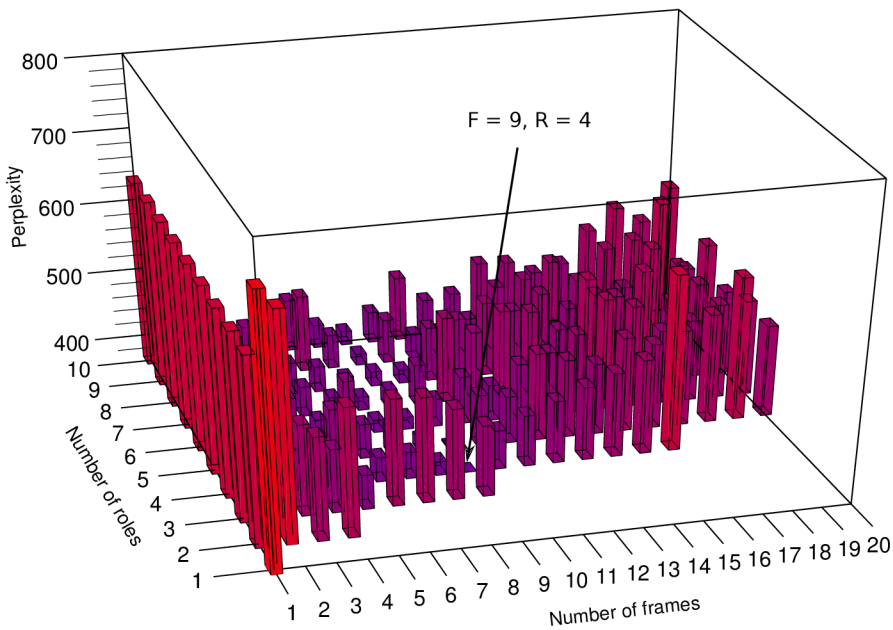


Figure 11.6: Perplexities for different values of F and R on the *synthetic* data set.

You can see that the lowest perplexity has been reached using the model with 9 semantic frames and 4 semantic roles. These are the same values as in the original data from which the *synthetic* data set has been generated. The second part of the convergence test on the *synthetic* data set has been performed using the non-parametric algorithm. It has been verified that the non-parametric algorithm had

converged to the same minimum perplexity and values of parameters in 43 iterations on average (10 independent runs of the algorithm).

Apart from validating the right number of frames and roles, what remains is to verify whether the assignment of semantic roles to corresponding semantic role realizations in the corpus is correct. Because the semantic role identifiers are not inferred in a deterministic manner, it is necessary to find the correct assignment of the semantic role labels from the original data to the inferred semantic roles identifiers. Formally, it is the **weighted maximum bipartite matching problem**, e.g. Diestel (2010), where the weight of the matching between semantic role label $l \in \{l_1, l_2, \ldots, l_R\}$ and semantic role identifier $i \in \{1, 2, \ldots, R\}$ is defined as the number of semantic role realizations they have in common.

weighted maximum bipartite matching problem

Finally, based on the mapping, we are able to measure the accuracy of the inferred semantic frames and their roles. The accuracy is defined as a fraction of the number of correctly assigned semantic roles and the number of all semantic role realizations in the corpus. The evaluation algorithms can be found in the `utils` folder, specifically in the Python script called `evaluateRandom.py`. The experiments showed that the perfect result, i.e. the accuracy equal to 1.0, had been reached after 29 iterations of the parametric LDA-Frames algorithm with hyperparameter estimation, and after 46 iterations of the non-parametric LDA-Frames algorithm with hyperparameter estimation. Both numbers are averages of 10 independent runs of the sampler.

11.5 SCALABILITY

The purpose of the experiments in this section is to investigate how well the parallelized implementation of the LDA-Frames algorithm performs in comparison to the sequential version. There are two aspects of the evaluation. First, it is necessary to evaluate the quality of the learned model measured by the perplexity of a test data set given the model. The second aspect is the scalability itself, i.e. how the learning time decreases with an increasing number of available processors. The primary data set for carrying out these experiments is *BNC-small*. This data set has been chosen to allow us to perform a large number of experiments in a reasonable time.

Figure 11.7 depicts the convergence of the parallelized LDA-Frames algorithm with hyperparameter estimation for 100 frames and 50 roles. The experiment has been carried out on a dual-processor server with two Intel Xeon X5675 CPUs, each with 6 physical cores running with a clock rate of 3.06 GHz. The sampler has been compiled using the g++ 4:4.8.1 compiler and run on the Ubuntu 13.10 operating system using 1, 5 and 10 cores. Although the parallelized algorithm is an approximation of the Collapsed Gibbs sampler for the LDA-Frames model, the figure shows that the parallelization has little

effect on the convergence and resulting quality. The little differences between the three curves are most likely caused by the stochastic nature of the algorithm.

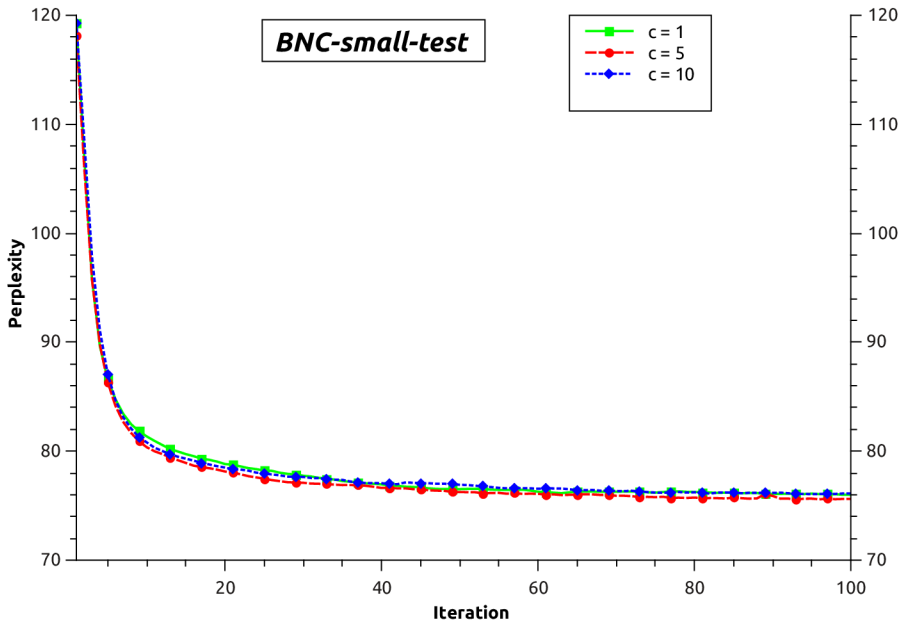


Figure 11.7: Convergence of the test perplexity for the parallelized LDA-Frames algorithm.

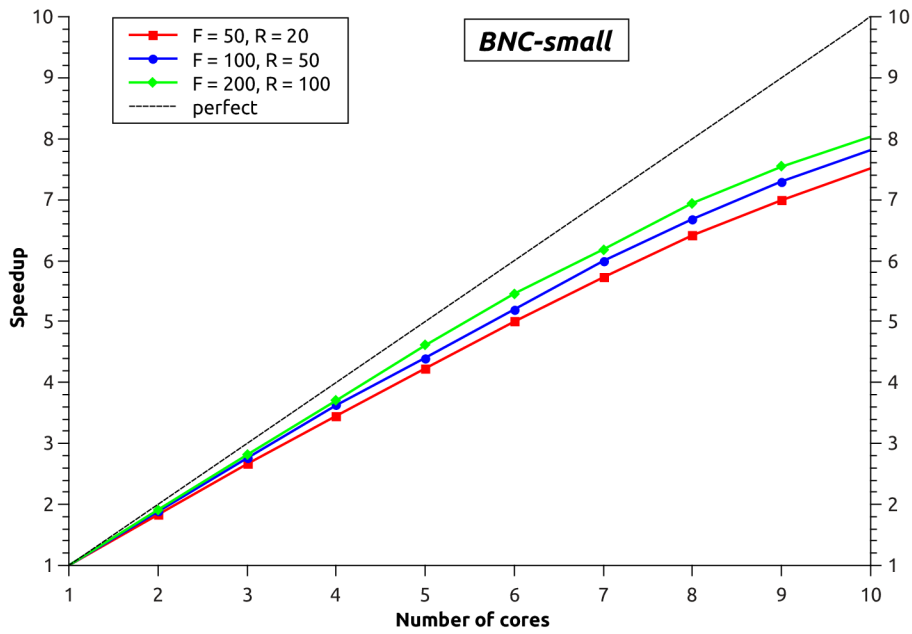


Figure 11.8: Speedup for 1 to 10 cores using the parallelized LDA-Frames algorithm.

The scalability of the parallel algorithm has been tested for the same parameter settings as in 11.4. For each configuration, 10 runs of the algorithm with the number of cores equal to $c \in \{1, 2, \ldots, 10\}$ have been performed. The running time has been measured using a standard Linux program `time`. The speedup is defined as the fraction of the sequential algorithm running time and the running time of the parallel version. Based on the results from 11.4, the number of iterations has been set to 100 for all experiments.

The results are shown in figure 11.8. You can see from the figure that the scalability of the parallel algorithm is slightly better for higher numbers of frames and roles. This behavior is caused by the fact that mutual exclusions in the program are less likely when the numbers of frames and roles are high.

11.6 COMPARISON WITH CPA

The previous experiments showed how well the LDA-Frames algorithm performs on small synthetic data. The goal of this section is to compare the probabilistic semantic frames inferred from the *BNC-big* data set with manually created frames developed within the Corpus Pattern Analysis project.

Corpus Pattern Analysis (Hanks and Pustejovsky, 2005) is a project started by Patrick W. Hanks, which aims at creating a mapping between verb meaning and its realizations in texts. The verb meanings are represented as verb patterns closely resembling semantic frames. The patterns are based on the Theory of Norms and Exploitations (Hanks, 2004), which is complementary to verb lexical units in Frame Semantics so that FrameNet provides an in-depth analysis of semantic frames, while CPA offers a systematic analysis of the patterns of meaning and use of each verb. The theory is currently being used to build Pattern Dictionary of English Verbs (PDEV)[6], which is intended to be a lexical database of verb usage patterns useful for students and teachers of English as well as a resource for practical NLP applications. One of the PDEV's strengths is that the dictionary is being built on the basis of corpus evidence from BNC, which makes the comparison with LDA-Frames learned from *BNC-big* reasonable.

The frames in PDEV (called patterns in CPA literature) are composed of clause roles described in the systemic grammar (Halliday, 1961). These roles assign a semantic interpretation to grammatical relations (typically subject and object) of a verb predicate, and can be seen as an equivalent of semantic roles defined in chapter 2.1. The meaning of the patterns in CPA is expressed as a set of implicatures. The words filling the subject and object relations can moreover be grouped into so called lexical sets (Jezek and Hanks, 2010).

6 http://deb.fi.muni.cz/cpa/

For example, one of the patterns for verb 'access' is

(11.2) `[[Human]]` **access** `[[Location]]`,

where 'access' is the verb predicate and 'Human' and 'Location' are semantic roles (called semantic types in CPA literature). The corresponding implicature is defined as

(11.3) *[[Human]] is able to enter and leave [[Location]].*

A screenshot of the complete entry for the verb 'access' taken on the PDEV web site is shown in figure 11.9. Apart from the pattern definitions and implicatures, the web provides the user with an access to corpus exemplifications of particular word meanings identified by pattern numbers.

access samplesize: **all=274** patterns: **4**

No.	%	Pattern / *implicatures*
1	<1%	**[[{Human = Computer User} | Computer 1]] access [[Information | Computer 2]]** *[[{Human = Computer User} | Computer 1]] obtains or retrieves [[Information]], typically from a database held on [[Computer 2]]*
2	<1%	**[[Human]] access [[Information = Neurological]]** *[[Human]] is able to obtain and retrieve [[Information = Neurological]], typically by thinking about it and remembering details*
3	<1%	**[[Human]] access [[Location]]** *[[Human]] is able to enter and leave [[Location]]*
4	<1%	**[[Human | Institution]] access [[Money]]** *[[Human | Institution]] is able to obtain and spend [[Money]] from a particular source requiring authorization*

Figure 11.9: Screenshot of the PDEV entry for verb 'access'.

The work of P. Hanks on PDEV has been followed by a project called Semantic Pattern Recognition[7], which has resulted in the Verb Pattern Sample (VPS) lexical resource (Cinková et al., 2012). Its main goal is to build a sample of the cleanest CPA data available for experiments, and explore its potential for an automatic lexical disambiguation. It is comprised of 30 English verbs from PDEV, which have been revised and cleaned. For each verb, there is a set of 300 randomly selected concordance samples. A subset of 50 samples have been analyzed by multiple annotators who assigned corresponding frame pattern identifiers to the concordances according to the similarity of implicatures. After the annotations had been finished, an inter-annotator agreement measured using Fleiss' kappa (Artstein and

7 http://ufal.mff.cuni.cz/project/spr/

Poesio, 2008) have been carried out. It provides us with valuable information about the complexity and ambiguity of the annotation process.

11.6.1 *Case Study: verbs 'access', 'submit' and 'pour'*

The evaluation of semantic frames automatically generated using the LDA-Frames algorithm against a CPA-based lexical resource is a two-folded problem. First, it is required to show how the generated semantic frames correspond with the verb patterns from CPA. The second task is an automatic assignment of semantic frames to corpus realizations. It can be seen as an automatic word sense disambiguation, where the assigned frame represents a particular word sense or one of the usage patterns of a single word sense.

Let us start with the former task. The goal is to show that the LDA-Frames algorithm is able to discover similar structures as the patterns from CPA. However, semantic types in CPA are represented by their labels, while semantic roles in LDA-Frames by probability distributions over words. Therefore, it is unclear how to link them automatically. That is the reason why we have selected three verbs and analyzed them manually. The verbs for this case study are 'access', 'submit' and 'pour'. These verbs have complex structures, are presented in both PDEV and VPS, and therefore have high quality annotations and are accompanied with information about the inter-annotator agreement.

In the beginning, it is necessary to transform CPA patterns to the format comparable with generated LDA-Frames data. Since we use the data inferred from BNC strictly for the subject and object grammatical relations, it is necessary (1) to omit intransitive patterns with no direct object, and (2) merge CPA patterns that differ only in semantic types other than those corresponding to subject or object. For example, pattern

(11.4) `[[Liquid]]` **pour** `[[NO OBJ]] [Adv[Direction]]`

of the verb 'pour' disallows object, and thus must be excluded from the analysis. Pattern merging, for instance, must be carried out for patterns

(11.5) `[[Human | Institution]]` **pour** `[[Money | Resource]]` (out | forth) (into [[Activity]] | into [[Entity]])

and

(11.6) `[[Human | Institution]]` **pour** `[[Money | Resource]]` down drain into black hole

where the semantic types for subject and object are identical. It is worth noticing that these limitations are not intrinsic to the LDA-Frames algorithm itself. It is able to handle both issues (empty objects and more grammatical relations than direct object and subject). The reason for using only the `subject-object` structures is that the experimentation with the same data as in previous sections is desirable. Intransitive verb patters, for example, are exemplified in the *synthetic* data set.

After the preprocessing of the CPA patterns, we get the following sets of frames (they are listed along with their implicatures):

ACCESS

(1) `[[Human = Computer User | Computer 1]]` **access** `[[Information | Computer 2]]`

[[Human = Computer User | Computer 1]] obtains or retrieves [[Information]], typically from a database held on [[Computer 2]]

(2) `[[Human]]` **access** `[[Information = Neurological]]`

[[Human]] is able to obtain and retrieve [[Information = Neurological]], typically by thinking about it and remembering details

(3) `[[Human]]` **access** `[[Location]]`

[[Human]] is able to enter and leave [[Location]]

(4) `[[Human | Institution]]` **access** `[[Money]]`

[[Human | Institution]] is able to obtain and spend [[Money]] from a particular source requiring authorization

SUBMIT

(1) `[[Human 1 | Institution 1]]` **submit** `[[Plan | Document]]`

[[Human 1 | Institution 1]] presents [[Plan | Document]] to [[Human 2 | Institution 2]] for approval

(2) `[[Human]]` **submit** `{claim | dispute}`

[[Human]] presents {claim | dispute} for a decision or approval by an authority

(3) `[[Human]]` **submit** `{[that-CLAUSE | QUOTE]}`

[[Human]] formally argues {[that-CLAUSE | QUOTE]} is the case

(4) `[[Human 1]]` **submit** `([[Self]])`

[[Human 1]] voluntarily undergoes or suffers [[Eventuality = Unpleasant]]

POUR

(1) `[[Human]]` **pour** `[[Liquid]]`

[[Human]] causes [[Liquid]] to flow from [[Physical Object = Container]] in a steady stream (into a container or onto a surface)

(2) `[[Human | Institution]]` **pour** `[[Money | Resource]]`

[[Human]] presents {claim | dispute} for a decision or approval by an authority

(3) `[[Human]]` **pour** `[[Emotion]] {out}`

[[Human]] expresses [[Emotion]] in an unrestrained way

(4) `[[Entity]]` **pour** `[[Stuff = Product]] {out}`

[[Entity]] produces [[Stuff = Product]] in large amounts

(5) `[Human]` **pour** `[{tear|tears}]`

[Human] weeps, [Human] sheds tears

For the comparison purposes, we have used the model inferred on *BNC-big* using the parallel LDA-Frames algorithm with hyperparameter estimation for a fixed number of frames and roles. The number of frames has been set to 1200 and the number of roles to 400. The model with this setting is operated on the project web site.

The matching between CPA and LDA-Frames has been carried out based on semantic interpretations of the semantic type labels and the implicatures at the CPA side, and based on the top ten realizations of semantic roles at the LDA-Frames side. Since there are possibly 1200 distinct semantic frames for each lexical unit in the LDA-Frames model, the set of candidates for matching has been limited to semantic frames with prior probability higher than 0.01. The matching is presented in table 11.4. The utilized semantic frames from the LDA-Frames model along with their top ten realizations are listed in appendix B.

lexical unit	CPA pattern	frame ID	prior prob.	subject role	object role
ACCESS	1	969	0.552632	245	245
	1	1133	0.151316	87	245
	2	1060	0.138158	219	53
	3	397	0.019737	193	20
	4	1051	0.013158	87	277
SUBMIT	1	670	0.440147	14	53
	1	453	0.230203	344	372
	2	819	0.018416	71	106
	4	918	0.022099	2	398

POUR	1	876	0.607226	219	120
	2	952	0.040793	193	277
	5	251	0.038462	225	225

Table 11.4: Matching between CPA and LDA-Frames.

The different granularity of CPA and LDA-Frames causes that some CPA patterns must be linked to multiple frames. For example, pattern (1) of 'access' is linked to frames 969 and 1113. They differ in the semantic roles for subject. While the subject slot in the CPA pattern is filled with both `Human` and `Computer` semantic types at the same time, these roles are treated separately in LDA-Frames. On the other hand, multiple semantic types in CPA are sometimes so specific that have a single equivalent in LDA-Frames. For example, both 'tear' and 'eye' are represented by one semantic role 225. Some idiomatic and rarely used patterns, for example pattern (3) of 'pour', have no equivalent in LDA-Frames with a prior probability higher than 0.01. Such patterns remained without a link to LDA-Frames. A statistic of the matching is summarized in table 11.5.

lexical unit	**patterns in PDEV**	**frames in LDA-Frames**	**patterns after merging**	**linked frames**
access	4	7	4	5
submit	6	12	4	4
pour	20	10	5	3

Table 11.5: Statistics of the matching between CPA and LDA-Frames.

The second and the third columns represent the number of patterns in PDEV and the number of semantic frames in LDA-Frames with the prior probability higher than 0.01, respectively. The column named 'patterns after merging' summarizes the number of patterns in PDEV after applying the merging and omitting rules as were described above. The last column represents the number of frames in LDA-Frames that were linked to PDEV patterns.

Having prepared the matching between CPA and LDA-Frames, we can now proceed to an automatic assignment of semantic frames to verbs in texts. The automatic annotation of syntactically parsed texts with LDA-Frames is very straightforward. The probabilistic character of LDA-Frames provides us with a natural way of choosing the most probable semantic frame for a lexical unit based on the semantic role realizations. Let u be the investigated lexical unit and $\mathbf{t} = t_1, t_2, \ldots, t_S$

semantic role realizations for slots $\{1, 2, \ldots, S\}$. Given the LDA-Frames model m, the probability of frame f is from 7.51 defined as

$$P(f|u, \mathbf{t}, m) = \frac{(fc_{f,u} + \alpha) \times \prod_{s=1}^{S} \frac{wc_{t_s, r_{f,s}} + \beta}{wc_{*, r_{f,s}} + V\beta}}{\sum_{i=1}^{F} \left[(fc_{i,u} + \alpha) \times \prod_{s=1}^{S} \frac{wc_{t_s, r_{i,s}} + \beta}{wc_{*, r_{i,s}} + V\beta} \right]}. \tag{11.4}$$

The most probable semantic frame f is then chosen as

$$f = \arg\max_{f'} P(f'|u, \mathbf{t}, m). \tag{11.5}$$

The evaluation of the automatic semantic frame assignment procedure has been carried out on manually adjudicated annotations from VPS. Unfortunately, the annotations do not contain annotations of semantic role realizations, which are required by the algorithm. This information has been automatically extracted using the Stanford parser and then manually verified by an annotator.

The quality of the automatic annotations is measured using three standard techniques – precision, recall and accuracy, defined as

$$\text{precision} = \frac{\text{correct}}{\text{correct} + \text{wrong}} \tag{11.6}$$

$$\text{recall} = \frac{\text{correct}}{\text{correct} + \text{skipped}}, \tag{11.7}$$

$$\text{accuracy} = \frac{\text{correct}}{\text{correct} + \text{wrong} + \text{skipped}}, \tag{11.8}$$

where correct is the number of correctly classified lexical units, wrong is the number of incorrectly classified lexical units, and skipped is the number of lexical units with known CPA pattern for which there is no equivalent in LDA-Frames.

The results from the comparison of CPA and LDA-Frames are presented in table 11.6.

lexical unit	**IAA**	**training examples**	**precision**	**recall**	**accuracy**
access	0.600	152	0.783	1.000	0.783
submit	0.764	550	0.939	0.838	0.795
pour	0.652	858	0.870	0.741	0.667

Table 11.6: Quality of the automatic assignment of LDA-Frames.

The second column represents the Fleiss' kappa values of the inter-annotator agreement on the data. It is worth emphasizing that the set of patterns for which the agreement was originally measured is slightly different from the set of merged patterns. The values, however, should correlate with the difficulty of annotations. The third column represents the number of realizations of particular lexical units in the training data. The last three columns summarize precision, recall and accuracy, respectively.

This case study for only three verbs is of course hardly representative. A more extensive study with equally detailed analysis, however, would require huge amounts of human annotator resources. To evaluate the results on a larger set of lexical units, we have performed a simplified analysis described in the following subsection.

11.6.2 *Comparison on 300 most frequent lexical units*

In order to test the quality of the generated frames on a larger set of lexical units, we have compared the generated LDA-Frames and CPA patterns for 300 most frequent lexical units from CPA. Because a corpus with annotated semantic type realizations for all 300 verbs is not available (the CPA corpus contains just frame identifiers without an annotation of corresponding arguments), we have only measured matching between the learned LDA-Frames data and the CPA patterns. It corresponds to the first task in the previous subsection. This simplified study is based on the experiment presented in Materna (2012a).

In contrast to the previous experiment where the frames have been linked manually for each lexical unit, here, we have linked the integer representations of semantic roles in LDA-Frames to semantic types in CPA. It is carried out without looking at particular frames or lexical units, and thus it requires less manual work (there are only 400 semantic roles in the LDA-Frames model). A disadvantage of this approach, however, is that the evaluation is not so accurate and produces slightly worse results. The mapping has been carried out manually so that at least one CPA role had to be assigned to a semantic role in LDA-Frames. Some of the roles have been assigned to more than one semantic role in CPA.

The correspondence has been measured as follows. First, we have measured the percentage of LDA-Frames that, after translating semantic role numbers to CPA semantic types, are presented in the set of CPA patterns for each lexical unit. Secondly, we have measured the percentage of lexical units, where the most frequent frame is same in CPA and LDA-Frames. The results are shown in table 11.7.

The results show that although the LDA-Frames patterns are created by a different method, LDA-Frames is an approach that is able to generate frame structures very similar to verb pattern

method	correspondence
All frames	52.8 %
Most frequent frames	61.8 %

Table 11.7: The correspondence of LDA-Frames and CPA patterns.

lexicons automatically. Moreover, it is able to automatically annotate syntactically parsed texts for the verbs from the case study.

11.7 SEMANTIC SIMILARITY OF LEXICAL UNITS

The semantic frames generated by the LDA-Frames algorithm are an interesting source of information about selectional preferences, but they can even be used for grouping semantically related lexical units. Individual semantic frames can hardly capture the whole semantic information about a lexical unit, nevertheless, the LDA-Frames approach provides information about the relatedness to every semantic frame we have inferred. After the inference process, each lexical unit u is connected with a probability distribution over semantic frames φ_u. Therefore, we can group lexical units with similar probability distributions together to make semantic clusters. We will call these clusters **similarity sets**. The ideas follow my recent work published in Materna (2012b).

similarity set

There are several methods for the comparison of probability distributions. We use the Hellinger distance (Hazewinkel, 2001), which measures the divergence of two probability distributions, and is a symmetric modification of the Kullback-Leibner divergence. For two probability distributions φ_a, φ_b, where $P(f|x)$ is the probability of frame f given lexical unit x, the Hellinger distance is defined as follows:

$$H(a,b) = \sqrt{\frac{1}{2}\sum_{f=1}^{F}\left(\sqrt{P(f|a)} - \sqrt{P(f|b)}\right)^2}. \tag{11.9}$$

Using the Hellinger distance, we can generate a ranked list of semantically similar words for every lexical unit u for which the semantic frames have been computed. Then the similarity set is chosen by selecting n best candidates or by selecting all candidates $c_1, c_2, \ldots, c_n$, where $H(u, c_i) < \tau$ for some threshold τ.

For the experimental purposes, we have selected all transitive verbs having their frequency in BNC greater than 100. The transitivity has been determined by looking into PDEV and selecting such verbs that have both subject valency and direct object valency presented. The constraints have been fulfilled by 4053 English verbs. These verbs have been enhanced with frame distributions acquired from the

LDA-Frames model trained on *BNC-big* with the hyperparameter estimation and the number of frames and roles set to 1200 and 400, respectively. Using the frame distributions, we have created a list of similar verbs for each lexical unit, sorted in ascending order based on the distance measure described above. The verbs with distance 1.0 were omitted. We have created 20 different thesauri from the sorted lists. For $1 \leqslant n \leqslant 20$, the thesaurus has been built as the set of at most first n items from the similarity lists for each lexical unit.

word sketch

The generated similarity sets have been compared with a competitive statistical thesaurus created using the Sketch Engine framework (Kilgarriff et al., 2004) from so called **word sketches**. Word sketches are automatic, corpus-based summaries of a word's grammatical and collocational behavior, which take as input a corpus of any language and corresponding grammar patterns. The resulting summaries are produced in the form of a ranked list of common word realizations for a grammatical relation of a given word. These statistics are compared in order to create lists of semantically similar lexical units. The Sketch Engine has been used to build 20 thesauri as in case of LDA-Frames from the same data. Because verbs with distance 1.0 have been omitted, not all lexical units have exactly n verbs in their similarity sets. Figure 11.10 shows the average number of verbs in the similarity sets for every n we have considered.

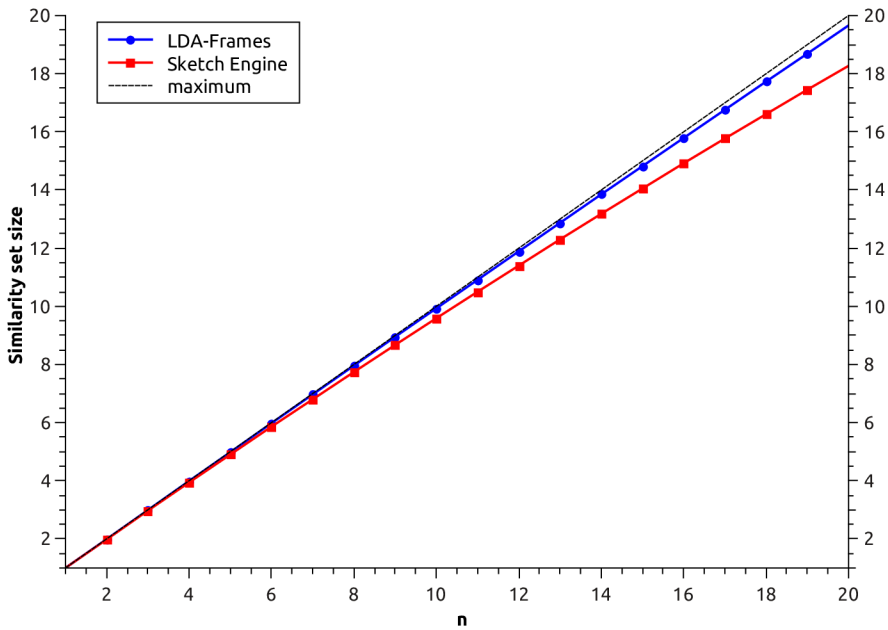


Figure 11.10: Average number of words in the similarity sets.

The results from the figure can be interpreted such that the Sketch Engine thesauri are stricter than the LDA-frames thesauri and produce smaller similarity sets.

In order to evaluate the quality of the generated thesauri, their similarity sets have been compared with synsets from the English WordNet 3.0. Since there are verbs that are included in the thesauri but are not included in WordNet (and vice versa), we have selected verbs contained in their intersection. It has resulted in a set of 2880 verbs. The quality has been measured as the average number of verbs from a similarity set contained in the corresponding WordNet synset, normalized by the size of the similarity set. Formally, let V be the number of investigated verbs, $T(v)$ the similarity set for verb v in thesaurus T and $W(v)$ the synset for verb v in WordNet. The score is then computed as

$$\text{Score}(T) = \frac{1}{V} \sum_{v=1}^{V} \frac{|T(v) \cap W(v)|}{|T(v)|}. \tag{11.10}$$

The resulting scores for similarity set sizes $1 \leqslant n \leqslant 20$ are shown in figure 11.11.

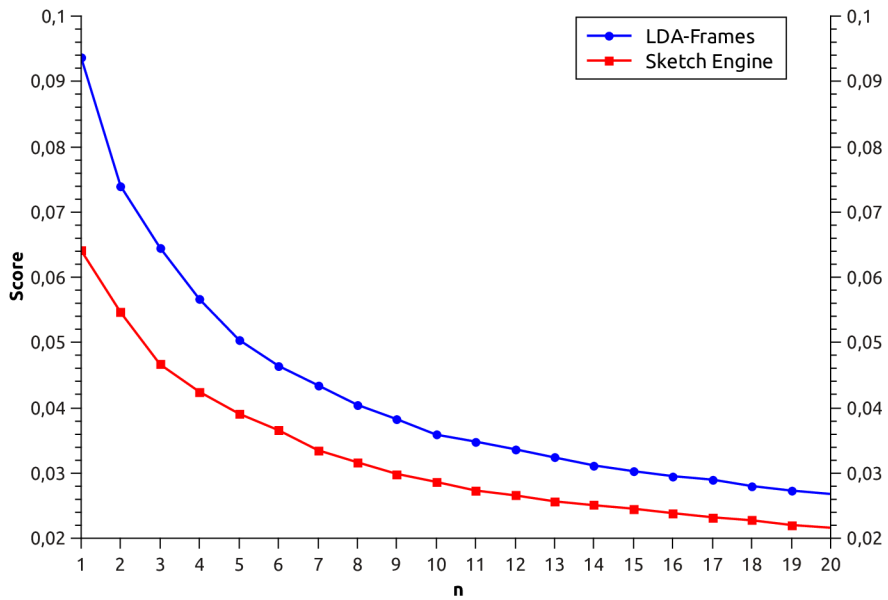


Figure 11.11: Comparison of the Sketch Engine thesauri with the LDA-frames thesauri.

It can be seen that the LDA-frames thesauri outperform the Sketch Engine for all choices of n. The difference is most noticeable when $n = 1$. This special case measures whether the most similar verb is presented in the corresponding WordNet synset. This condition is satisfied in approximately 9.5 % verbs for LDA-frames and 6.5 % for Sketch Engine. The scores may seem to be small but it is worth mentioning that many verbs from the synsets are excluded from the evaluation (they were outside the intersection of the lexical units),

thus the theoretical maximum is deeply below 100 %. Moreover, only subject and object grammatical relations have been taken into consideration when computing the similarity of lexical units. This means, for instance, that English verbs 'eat' and 'cook' have very high similarity scores, because they both are used in the same contexts, and thus have completely identical semantic frames. It is apparent that the algorithm would achieve better results if there were used more than two grammatical relations. Specifically, the verbs 'eat' and 'cook' could be differentiated, for example, by adding a grammatical relation corresponding with the instrument for eating or cooking.

For further experiments, the resulting LDA-Frames thesaurus is available on the LDA-Frames project web site.

12 CONCLUSION

The objective of this thesis was to explore the state of the art in the field of electronic valency lexicons and semantic frames, and to propose a method for creating such electronic resources automatically. As was shown at the beginning, there are many frame-based electronic lexicons, but they are mostly created manually, which requires huge amounts of human manual work. Moreover, the lexicons rarely provide statistics about their entries supported by corpus evidence. Such statistics are very useful and desired in automatic natural language processing tasks.

As a reaction to the current situation, this thesis offers an automatic method for creating probabilistic semantic frames called LDA-Frames. The method requires a syntactically parsed corpus of any language as an input, and for each lexical unit from the corpus, it produces a probability distribution over the set of generated semantic frames. The semantic frames consist of semantic roles represented as probability distributions over their realizations. The whole model is described as a generative probabilistic graphical model. The inference algorithm for the model is based on the collapsed Gibbs sampling, and generates the result in a completely unsupervised manner.

Apart from the training corpus, the algorithm requires an input of two parameters – the number of roles and the number of frames, and several hyperparameters controlling shapes of the distributions. Since these parameters depend on the provided corpus and their choice may not be obvious, this thesis describes a non-parametric version of LDA-Frames, which is an extension of the original model that estimates the parameters automatically. This thesis also proposes an automatic procedure for estimating the hyperparameters.

The quality of the inferred semantic frames depends on the size of the training data. The bigger the data is, the more information is available to build a high quality lexical resource. Large data sets, however, require long running time of the algorithm. To overcome this difficulty, we have proposed a distributed algorithm that enables the inference procedure to run in a parallel manner. It has been shown that the parallelization has almost no impact on the quality of the frames and that the scalability is pretty good for the number of physical processors ranging from 1 to 10.

A standard method of evaluating probabilistic models is to measure the probability of unseen data given the model. We have created training and testing data sets from the British National Corpus automatically parsed using the Stanford parser, and shown that

the inference algorithm successfully converges to optimal values of perplexity in both parametric and non-parametric models. Moreover, no overfitting has been observed. Because the convergence itself does not guarantee meaningfulness of the model, we have shown on a small data set that it is able to perfectly reconstruct synthetically prepared semantic frames.

The range of possible utilizations of the proposed models in NLP is wide. The most straightforward applications are word sense induction and word sense disambiguation. The applicability of LDA-Frames to these tasks have been shown in a case study comparing LDA-Frames with the CPA project. The study has demonstrated that the induced semantic frames are very similar to manually created patterns from CPA, and that the probabilistic manner of LDA-Frames enables to semantically annotate unseen text with high precision. Another application of LDA-Frames is its utilization for the creation of thesauri. It has been presented that the thesaurus built from an LDA-Frames model reaches a higher compliance with manually compiled WordNet synsets than a competitive approach based on the Sketch Engine.

To make experimenting with LDA-Frames comfortable, the probabilistic semantic frames and the thesaurus computed on the British National Corpus are available on-line on the project's home page: `http://nlp.fi.muni.cz/projekty/lda-frames`. Source codes of the inference algorithm and other auxiliary scripts are accessible from the home page as well.

The LDA-Frames approach is even applicable outside the natural language processing field. Its statistical nature enables a discovery of hidden patterns in any types of data that can be parsed into tuples of objects. Such applications, however, remain for future work.

Part III

APPENDIX

A

COMPILATION AND USAGE INSTRUCTIONS

A.1 INTRODUCTION

LDA-frames is an approach to identifying semantic frames from semantically unlabeled text corpora. There are many frame formalisms but most of them suffer from the problem that all the frames must be created manually and the set of semantic roles must be predefined. The LDA-Frames approach, based on the Latent Dirichlet Allocation, avoids both these problems by employing statistics on a syntactically annotated corpus. The only information that must be given is a number of semantic frames and a number of semantic roles. This limitation, however, can be avoided by an automatic estimation of both these parameters. The model which is able to estimate the number of frames and roles automatically is called Non-parametric LDA-Frames.

A.2 DEPENDENCIES

- Python >= 2.5
- Numpy >= 1.6
- NLTK >= 2.0
- Scipy >= 0.9
- Java 6 >= JDK 1.6 (Only when using the Stanford parser for generating the syntactic dependencies.)
- g++ >= 4.6
- Boost program options development package >= 1.46
- GNU Scientific Library (GSL) development package >= 1.15
- OpenMP (Only when using the multi-threaded sampler)

A.3 BUILDING BINARIES FROM THE SOURCE CODES ON UBUNTU 13.10.

Install necessary dependencies.

```
$ sudo apt-get install g++ libboost-program-options-dev \
                       libgsl0-dev python-numpy python-nltk \
                       python-scipy
```

Move into the working directory of the sampler.

```
$ cd sampler
```

Build the binary file.

```
$ make
```

A.4 EXAMPLE OF THE USAGE

Generate input data for the sampler along with a dictionary.

```
$ preprocessing/generate_samplerinput.py \
    -l data/synthetic/synthetic.rel data/synthetic/
```

Run the Non-Parametric LDA-Frames sampler with 100 iterations.

```
$ sampler/ldaframes-sampler --seed=1 \
    -I 100 data/synthetic/train.dat data/synthetic/
```

Evaluate the produced frames against the original data.

```
$ utils/evaluateRandom.py data/synthetic/synthetic.rel.chck \
    data/synthetic/
```

Generate frame distributions for lexical units and word distributions for semantic roles.

```
$ preprocessing/generate_distributions.py data/synthetic/
```

Generate similarity matrix for all lexical units (based on the comparison of their frame distributions).

```
$ preprocessing/generate_similarities.py data/synthetic/
```

Show generated semantic frames for lexical unit "eat".

```
$ libldaf/showFrames.py -r 4 data/synthetic/ eat
```

B

LDA-FRAMES MODEL – SELECTED FRAMES

frame	subject realizations	object realizations
969	system user application datum program information software computer version module	system user application datum program information software computer version module
1133	person people patient one child woman country company government who	system user application datum program information software computer version module
1060	person people man who one child woman family father mother	information report letter copy message number detail statement list document
397	person company government firm bank authority group people who council	service system work range area program course market one scheme
1051	person people patient one child woman country company government who	cost price money rate $ million amount number share %
670	person government group student company people member party who authority	information report letter copy message number detail statement list document
453	person government council party committee authority minister board parliament commission	plan agreement policy proposal bill contract resolution decision program law
819	person man herself friend mother people story father other one	claim offer appeal charge case invitation argument suggestion idea report
918	person one man people who someone woman mother nobody father	himself herself myself yourself ourselves body feeling hellip yourselve time
876	person people man who one child woman family father mother	coffee tea water wine glass drink cup beer sigh whisky

952	person company government firm bank authority group people who council	cost price money rate $ million amount number share %
251	face light smile eye tear shadow sun cheek look blood	face light smile eye tear shadow sun cheek look blood

Table B.1: Top-ten realizations of semantic roles in LDA-Frames.

BIBLIOGRAPHY

David J. Aldous. Exchangeability and Related Topics. *École d'Été de Probabilités de Saint-Flour XIII – 1983*, 1117:1 – 198, 1985.

Charles E. Antoniak. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *Annals of Statistics*, 2(6), November 1974.

Ron Artstein and Massimo Poesio. Inter-coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596, December 2008. ISSN 0891-2017.

Satanjeev Banerjee and Ted Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 136–145, London, UK, 2002. Springer-Verlag. ISBN 3-540-43219-1.

Carol A. Bean and Rebecca Green. *Relationships in the Organization of Knowledge*. Springer, 2001. ISBN 978-0792368137.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. ISBN 978-0387310732.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993 – 1022, 2003.

Donald M. Borchert. *Encyclopedia Of Philosophy*. Macmillan Reference USA; 2 edition, 2005. ISBN 978-0028657905.

George E. P. Box. *Empirical Model-Building and Response Surfaces*. Wiley; 1 edition, 1987. ISBN 978-0471810339.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24 – 42, 2002.

Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. The Salsa Corpus: A German Corpus Resource for Lexical Semantics. In *The International Conference on Language Resources and Evaluation (LREC)*. Genoa, Italy, 2006.

Rainer E. Burkard and Eranda Çela. Linear Assignment Problems and Extensions. *Handbook of Combinatorial Optimization, Supplement Volume A*, 1999.

Kenneth P. Burnham and David R. Anderson. *Model Selection and Multi-Model Inference: A Practical Information-Theoretic Approach*. Springer, 2 edition, 2002. ISBN 0387953647.

Noam Chomsky. *Aspects of the Theory of Syntax*. The MIT Press, 1969a. ISBN 978-0262530071.

Noam Chomsky. Some Empirical Assumptions in Modern Philosophy of Language. *Philosophy, Science and Method: Essays in Honor or Ernest Nagel*, 1969b.

Noam Chomsky. *Lectures on Government and Binding: The Pisa Lectures*. Mouton de Gruyter; 7 Sub edition, 1993. ISBN 978-3110141313.

Silvie Cinková, Martin Holub, Adam Rambousek, and Lenka Smejkalová. A Database of Semantic Clusters of Verb Usages. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, LREC 2012, pages 3176–3183. European Language Resources Association, 2012. ISBN 978-2-9517408-7-7.

Gregory F. Cooper. Probabilistic Inference Using Belief Networks Is NP-Hard. *Artificial Intelligence*, 42:309–347, 1990.

D. Alan Cruse. *Lexical Semantics*. Cambridge University Press, 1986. ISBN 978-0521276436.

Hoa Trang Dang. *Investigations Into the Role of Lexical Semantics in Word Sense Disambiguation*. PhD thesis, University of Pennsylvania, 2004.

Hoa Trang Dang, Karin Kipper, Martha Palmer, and Joseph Rosenzweig. Investigating Regular Sense Extensions Based on Intersective Levin Classes. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 293–299, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

Marie-Catherine de Marneffe and Christopher D. Manning. Stanford Typed Dependencies Manual. Technical report, Stanford University, 2008.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating Typed Dependency Parses from Phrase Structure Parses. In *The International Conference on Language Resources and Evaluation (LREC) 2006*, 2006.

Reinhard Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer, 2010. ISBN 978-3642142789.

Robert M. W. Dixon. *A New Approach to English Grammar on Semantic Principles*. Clarendon Press, Oxford, 1991. ISBN 978-0198242727.

David R. Dowty. Thematic Roles and Semantics. In *Proceedings of the Twelfth Annual Meeting of the Berkeley Linguistics Society*, pages 340–354, 1986.

S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using Latent Semantic Analysis to Improve Access to Textual Information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, pages 281–285, New York, NY, USA, 1988. ACM. ISBN 0-201-14237-6.

Benjamin Van Durme and Daniel Gildea. Topic models for corpus-centric knowledge generalization. Technical report, Computer Science Department, The University of Rochester, Rochester, NY, 2009.

Carl Eckart and Gale Young. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika*, 1(3):211–218, 1936. ISSN 0033-3123.

Roger Eckhardt. Stan Ulam, John von Neumann, and the Monte Carlo Method. *Los Alamos Science*, pages 131–137, 1987.

Elena Erosheva, Stephen Fienberg, and John Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101(suppl. 1):5220–5227, 2004.

Michael D. Escobar and Mike West. Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998. ISBN 978-0-262-06197-1.

Thomas S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1:209 – 230, 1973.

Charles J. Fillmore. The Case for Case. In *Universals in Linguistic Theory*. New York: Holt, Rinehart and Winston, 1968.

Charles J. Fillmore. Topics in Lexical Semantics. In *Current Issues in Linguistic Theory*. Indiana University Press, 1976.

Charles J. Fillmore. Scenes-and-frames Semantics. In *A. Zampolli, ed. Linguistic Structures Processing*, pages 55 – 81. Amsterdam, North-Holland, 1977a.

Charles J. Fillmore. The Case for Case Reopened. In *Syntax and Semantics: Grammatical Relations*. Academic Press Inc., 1977b.

Charles J. Fillmore. Frame Semantics. In *Linguistics in the Morning Calm*, pages 111 – 137. Seoul, Hanshin Publishing Co., 1982.

Charles J Fillmore, Christopher R Johnson, and Miriam R L Petruck. Background to FrameNet. In *International Journal of Lexicography*, pages 235 – 250. Oxford University Press, 2003.

John Rupert Firth. A Synopsis of Linguistic Theory 1930–1955. *Studies in linguistic analysis*, 51:1–31, 1957.

Bela A. Frigyik, Amol Kapila, and Maya R. Gupta. Introduction to the Dirichlet Distribution and Related Processes. Technical report, University of Washington, December 2010.

G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum. Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '88, pages 465–480, New York, NY, USA, 1988. ACM. ISBN 2-7061-0309-4.

Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, Ro Oltramari, and Luc Schneider. Sweetening Ontologies with DOLCE. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, EKAW '02, pages 166 – 181, London, UK, UK, 2002. Springer. ISBN 3-540-44268-5.

Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–472, 1992. ISSN 08834237.

Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, nov 1984. ISSN 0162-8828.

John Geweke. Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments. In *Bayesian Statistics*, pages 169–193. University Press, 1992.

Charles J. Geyer. Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4):473–483, 1992. ISSN 08834237.

Daniel Gildea and Daniel Jurafsky. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288, September 2002. ISSN 0891-2017.

Mark Girolami and Ata Kabán. On an equivalence between PLSI and LDA. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 433–434, New York, NY, USA, 2003. ACM. ISBN 1-58113-646-3.

Talmy Givón. *Syntax: a Functional-Typological Introduction*. J. Benjamins Pub. Co, 1990. ISBN 9781556190964.

Trond Grenager and Christopher D. Manning. Unsupervised Discovery of a Statistical Verb Lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 1–8, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. ISBN 1-932432-73-6.

Thomas L. Griffiths and Mark Steyvers. Finding Scientific Topics. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 5228 – 5235, 2004.

Jeffrey Steven Gruber. *Studies in Lexical Relations*. PhD thesis, Massachusetts Institute of Technology, Department of Modern Languages, 1965.

Liliane Haegeman. *Introduction to Government and Binding Theory*. Blackwell Publishers Inc, Malden, Massachusetts, USA, 1994. ISBN 0631190678.

Jan Hajič. Complex Corpus Annotation: The Prague Dependency Treebank. In *Insight into Slovak and Czech Corpus Linguistics*. Veda, Vydavateĺstvo SAV, Bratislava, 2005.

Michael Halliday. Categories of the theory of grammar. *Word*, page 241–92, 1961.

Patrick Hanks. The Syntagmatics of Metaphor and Idioms. In *International Journal of Lexicography*, pages 245 – 274. Oxford University Press, 2004.

Patrick Hanks and James Pustejovsky. A Pattern Dictionary for Natural Language Processing. In *Revue Francaise de Langue Appliquée*. Brandeis University, 2005.

W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, apr 1970. ISSN 1464-3510.

Michiel Hazewinkel. *Encyclopedia of Mathematics*. Springer, 2001. ISBN ISBN 978-1-55608-010-4.

Ohara Kyoko Hirose, Seiko Fujii, Toshio Ohori, Ryoko Suzuki, Hiroaki Saito, and Shun Ishizaki. The Japanese FrameNet Project: An Introduction. In *The Fourth international conference on Language Resources and Evaluation (LREC'04)*, pages 9 – 11, 2004.

Dana Hlaváčková and Aleš Horák. VerbaLex – New Comprehensive Lexicon of Verb Valencies for Czech. In *Computer Treatment of Slavic and East European, Languages Bratislava*, pages 107 – 115. Slovakia: Slovenský národný korpus, 2006.

Thomas Hofmann. Probabilistic Latent Semantic Analysis. In *Proceedings of Uncertainty in Artificial Intelligence, UAI*, 1999.

Nancy Ide and Jean Véronis. Introduction to the Special Issue on Word Sense Disambiguation: the State of the Art. *Computational Linguistics*, 24(1):2–40, 1998. ISSN 0891-2017.

Ray S. Jackendoff. *Semantic Interpretation in Generative Grammar*. The MIT Press, 1974. ISBN 978-0262600071.

Ray S. Jackendoff. *Semantics and Cognition*. The MIT Press, 1985. ISBN 978-0262600132.

Elisabetta Jezek and Patrick Hanks. What Lexical Sets Tell us about Conceptual Categories. *Lexis*, 4:7–22, 2010. ISSN 1951-6215.

Norman L. Johnson and Samuel Kotz. *Urn Models and Their Applications: An Approach to Modern Discrete Probability Theory*. John Wiley and Sons Inc., New York, 1977. ISBN 978-0471446309.

Adam Kilgarriff, Pavel Rychlý, Pavel Smrž, and David Tugwell. The Sketch Engine. In *Proceedings of the Eleventh EURALEX International Congress*, pages 205–116. Lorient, France, 2004.

Jin H. Kim and Judea Pearl. A Computational Model for Causal and Diagnostic Reasoning in Inference Systems. In *Proceedings of the Eighth international joint conference on Artificial intelligence - Volume 1*, IJCAI'83, pages 190–193, San Francisco, CA, USA, 1983. Morgan Kaufmann Publishers Inc.

Paul Kingsbury and Martha Palmer. From TreeBank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, 2002.

Paul Kingsbury, Martha Palmer, and Mitch Marcus. Adding Predicate Argument Structure to the Penn TreeBank. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 252–256, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. Class-Based Construction of a Verb Lexicon. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 691–696. AAAI Press, 2000. ISBN 0-262-51112-6.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. A Large-Scale Classification of English Verbs. *Language Resources and Evaluation*, 42(1):21–40, 2008.

S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

Dan Klein and Christopher D. Manning. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press; 1 edition, 2009. ISBN 978-0262013192.

Sopan Govind Kolte and Sunil G. Bhirud. Word Sense Disambiguation Using WordNet Domains. In *ICETET '08 Proceedings of the 2008 First International Conference on Emerging Trends in Engineering and Technology*, pages 1187–1191, 2008. ISBN 978-0-7695-3267-7.

Henry Kucera, W. Nelson Francis, John B. Carroll, and W. F. Twaddell. *Computational Analysis of Present Day American English*. Brown University Press; 1St Edition, 1967. ISBN 978-0870571053.

Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

P. J. M. Laarhoven and E. H. L. Aarts, editors. *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, Norwell, MA, USA, 1987. ISBN 9-027-72513-6.

Joel Lang and Mirella Lapata. Unsupervised Induction of Semantic Roles. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 939–947, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5.

Joel Lang and Mirella Lapata. Unsupervised Semantic Role Induction via Split-merge Clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1117–1126, Stroudsburg, PA, USA, 2011a. Association for Computational Linguistics. ISBN 978-1-932432-87-9.

Joel Lang and Mirella Lapata. Unsupervised Semantic Role Induction with Graph Partitioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1320–1331, Stroudsburg, PA, USA, 2011b. Association for Computational Linguistics. ISBN 978-1-937284-11-4.

Douglas Lenat. CYC: A Large-Scale Investment in Knowledge Infrastructure. In *Communications of the ACM*, volume 38, pages 33 – 38, 1995.

Beth Levin. *English Verb Classes and Alternation: A Preliminary Investigation*. The University of Chicago Press, 1993.

Xiaobin Li, Stan Szpakowicz, and Stan Matwin. A WordNet-based Algorithm for Word Sense Disambiguation. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1368–1374, 1995.

Jun S. Liu. The collapsed Gibbs sampler in Bayesian computations with application to a gene regulation problem. *Journal of the American Statistical Association*, 89(958–966), 1994.

Jun S. Liu, W. H. Wong, and A. Kong. Covariance Structure of the Gibbs Sampler with Applications to the Comparisons of Estimators and Augmentation Schemes. *Biometrika*, 81(1):27–40, 1994.

Markéta Lopatková, Zdeněk Žabokrtský, and Karolína Skwarska. Valency Lexicon of Czech Verbs: Alternation-based Model. In *The International Conference on Language Resources and Evaluation (LREC)*. Genoa, Italy, 2006.

John Lyons. *Semantics: Volume 1*. Cambridge University Press; 1st edition, 1977a. ISBN 978-0521291866.

John Lyons. *Semantics: Volume 2*. Cambridge University Press; 1st edition, 1977b. ISBN 978-0521291866.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press; 1 edition, 1999. ISBN 978-0262133609.

Jiří Materna. LDA-Frames: An Unsupervised Approach to Generating Semantic Frames. In *Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I*, CICLing'12, pages 376–387, Berlin, Heidelberg, 2012a. Springer-Verlag. ISBN 978-3-642-28603-2.

Jiří Materna. Building A Thesaurus Using LDA-Frames. In *6th Workshop on Recent Advances in Slavonic Natural Language Processing*, pages 97 –103. Tribun EU, 2012b. ISBN 978-80-263-0313-8.

Jiří Materna. Parameter Estimation for LDA-Frames. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 482–486, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

Paola Merlo and Eva Esteve Ferrer. The Notion of Argument in Prepositional Phrase Attachment. *Computational Linguistics*, 32(3): 341–378, 2006. ISSN 0891-2017.

Nicholas Metropolis and Stanislaw M. Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341, sep 1949. ISSN 01621459.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. ISSN 00219606.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Five Papers on WordNet. Technical report, Princeton University, 1993.

Thomas P. Minka. Estimating a Dirichlet distribution. Technical report, Microsoft Research, 2009.

Marvin Minsky. A Framework for Representing Knowledge. In *Psychology of Computer Vision*. McGraw-Hill, New York, 1975.

Ashutosh Modi, Ivan Titov, and Alexandre Klementiev. Unsupervised Induction of Frame-semantic Representations. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, WILS '12, pages 1–7, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Roberto Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, February 2009. ISSN 0360-0300.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed Algorithms for Topic Models. *The Journal of Machine Learning Research*, 10:1801–1828, December 2009. ISSN 1532-4435.

Mark Newman and Gerard T. Barkema. *Monte Carlo Methods in Statistical Physics*. Oxford University Press, 1999. ISBN 0198517963.

I. Niles and A. Pease. Towards a Standard Upper Ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, pages 2 – 9, 2001.

Peter Norvig, Alon Halevy, and Fernando Pereira. The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 24(2):8–12, mar 2009. ISSN 1541-1672.

Klára Osolsobě and Karel Pala. Czech Stem Dictionary. In *Sborník prací filozofické fakulty brněnské univerzity*, pages 51 – 60. Masarykova Univerzita, Brno, 1993.

Karel Pala and Pavel Smrž. Building Czech WordNet. In *Romanian Journal of Information Science and Technology*, volume 7, pages 1–2. Romanian Academy, 2004.

Karel Pala and Pavel Ševeček. Valence českých sloves. In *Sborník prací FFBU*, pages 41 – 54. Masarykova Univerzita, Brno, 1997.

Giorgio Parisi. *Statistical Field Theory*. Westview Press, 1998. ISBN 978-0738200514.

Charles Sanders Peirce. *Collected Papers*. Harvard University Press, 1931 – 1958. ISBN 978-0674138001.

Steven T. Piantadosi, Harry Tily, and Edward Gibson. The Communicative Function of Ambiguity in Language. *Cognition*, 2011.

Jim Pitman. Poisson&Ndash;Dirichlet and GEM Invariant Distributions for Split-and-Merge Transformations of an Interval Partition. *Combinatorics, Probability and Computing*, 11(5):501–514, 2002. ISSN 0963-5483.

Alexandrin Popescul, Lyle H. Ungar, David M. Pennock, and Steve Lawrence. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 437–444, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-800-1.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. Semantic Role Labeling Using Different Syntactic Views. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL 2005, pages 581–588, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. *A Comprehensive Grammar of the English Language*. Longman; 2nd Revised edition, 1985. ISBN 978-0582517349.

Andrew Radford. *Transformational Grammar*. Cambridge University Press, Cambridge, MA, USA, 1988. ISBN 0521 34750 5.

Adrian E. Raftery and Steven Lewis. How Many Iterations in the Gibbs Sampler? In *Bayesian Statistics 4*, pages 763–773. Oxford University Press, 1992.

Howard Raïffa and Robert Schlaifer. *Applied Statistical Decision Theory*. Wiley classics library. Wiley, 2000. ISBN 9780471383499.

Alan Ritter, Mausam, and Oren Etzioni. A Latent Dirichlet Allocation Method for Selectional Preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 424–434, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387212396.

G. Ronning. Maximum Likelihood Estimation of Dirichlet Distributions. *Journal of Statistical Computation and Simulation*, 32 (4):215–221, 1989.

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. Inducing a Semantically Annotated Lexicon via EM-based Clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 104–111, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. ISBN 1-55860-609-3.

J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, and J. Scheffczyk. FrameNet II: Extended Theory and Practice, 2006. http://www.icsi.berkeley.edu/framenet.

Joe Sachs. *Aristotle's Metaphysics*. Green Lion Press, Santa Fe, New Mexico, 1999. ISBN 1-888009-03-9.

John I. Saeed. *Semantics*. Wiley-Blackwell, 3 edition, 2009. ISBN 978-1405156394.

Karin Kipper Schuler. *VerbNet: a Broad-Coverage, Comprehensive Verb Lexicon*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 2005.

Diarmuid Ó. Séaghdha. Latent Variable Models of Selectional Preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 435–444, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Jayaram Sethuraman. A Constructive Definition of Dirichlet Priors. *Statistica Sinica*, 4(2):639–650, 1994.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. D. Reidel Publishing Company, Dordrecht, 1986.

Andrzej Siemiński. WordNet Based Word Sense Disambiguation. In *Proceedings of the Third international conference on Computational collective intelligence: technologies and applications - Volume Part II*, ICCCI'11, pages 405–414. Springer-Verlag, 2011. ISBN 978-3-642-23937-3.

John Simpson and Edmund Weiner. *The Oxford English Dictionary*. Clarendon Press; 2nd Edition Edition, 1989. ISBN 978-0870571053.

Mark Steyvers and Tom Griffiths. *Probabilistic Topic Models*. Lawrence Erlbaum Associates, 2007. ISBN 1410615340.

Carlos Subirats and Miriam Petruck. Surprise: Spanish Framenet. In *International Congress of Linguists. Workshop on Frame Semantics*. Prague, Matfyzpress, 2003.

Robert S. Swier and Suzanne Stevenson. Unsupervised Semantic Role Labelling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, 2004.

Yee W. Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes . *Journal of the American Statistical Association*, 101:1566 – 1581, 2006.

Lucien Tesnière. *Éléments de syntaxe structurale*. Klincksieck, Paris, 1959. ISBN 2252018615.

Ivan Titov and Alexandre Klementiev. A Bayesian Approach to Unsupervised Semantic Role Induction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 12–22, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. ISBN 978-1-937284-19-0.

Piek Vossen and Graeme Hirst. *EuroWordNet: A Multilingual Database with Lexical Semantic*. Kluwer Academic Publisher, 1998.

Zdeněk Žabokrtský. *Valency Lexicon of Czech Verbs*. PhD thesis, Charles University, Prague, 2005.

Bruce Walsh. Markov Chain Monte Carlo and Gibbs Sampling. Lecture Notes for EEB 581, version 26, 2004.

Nicolas Wicker, Jean Muller, Ravi Kiran Reddy Kalathur, and Olivier Poch. A Maximum Likelihood Approximation Method for Dirichlet's Parameter Estimation. *Computational Statistics and Data Analysis*, 52(3):1315–1322, January 2008. ISSN 0167-9473.

Ludwig Lazarus Zamenhof. *Fundamento de Esperanto*. Edistudio, Italy; First Edition edition, 1991. ISBN 978-8870360462.

INDEX

acyclic, 45
antonymy, 7
aperiodicity, 53
Aspect Model, 71

bag-of-words representation, 69
base distribution, 92
Bayesian network, 41
Bernoulli distribution, 60
beta distribution, 61
beta function, 112
binomial coefficient, 60
binomial distribution, 60
burn-in period, 56

calculus of variations, 49
categorical distribution, 61
Chapman-Kolomogrov equation, 53
child frame, 33
Chinese restaurant franchise, 99
Chinese restaurant process, 91
classification, 45
collapsed Gibbs sampler, 58
compositional semantics, 5
conditional independence, 42
conjugate prior, 58, 59
core frame element, 32
cycle, 45

deep case, 31
dependent, 124
digamma function, 111
directed acyclic graph, 45
Dirichlet distribution, 63
Dirichlet process, 91
Dirichlet-Multinomial distribution, 110
discriminative model, 45
domain-specific ontology, 8

exponential distribution, 65

extra-thematic frame element, 32

factor, 42
frame, 9
frame element, 31, 32
frame semantics, 31
functional, 49
functional generative description, 36

gamma distribution, 64, 113
gamma function, 61
GEM distribution, 95
generative model, 45
Gibbs sampler, 57
governor, 124
graph edge, 41
graph node, 41

head, 124
hidden variable, 44
Hierarchical Dirichlet process, 98
holonymy, 7
homonymy, 5
hypernymy, 7
hyponymy, 7

irreducibility, 53

Kullback-Leibler divergence, 49

Latent Dirichlet Allocation, 68, 74
Latent Semantic Analysis, 68
Latent Semantic Indexing, 67
latent variable, 44
lexical semantics, 5
lexical unit, 5, 31
lexicon, 5
logistic regression, 46

Markov chain, 53

Markov chain Monte Carlo, 48
Markov property, 53
Markov random field, 41
mean, 43
Mean-Field approximation, 50
meronymy, 7
Metropolis-Hasting algorithm, 53
Monte Carlo, 50
multinomial distribution, 61

naïve Bayes classifier, 46
near synonymy, 6
non-parametric distribution, 59
normal distribution, 43

observed variable, 44

parametric distribution, 59
parent frame, 33
peripheral frame element, 32
perplexity, 131
plate notation, 43
polysemy, 5
precision, 43
prepositional phrase attachment, 17
probabilistic graphical model, 41
Probabilistic Latent Semantic Analysis, 68
proposal distribution, 52

rate parameter, 64
regression, 45
rejection sampling, 52
reversibility condition, 54

self-conjugate prior, 59
semantic frame, 5, 22, 31
semantic role, 10
semantic role labeling, 21
semantic type, 34
shape parameter, 64
similarity set, 145
simplex, 63
simulated annealing, 56
Singular Value Decomposition, 68
slot, 9
standard deviation, 43
stationary distribution, 54
statistical induction, 45
statistical inference, 45, 48
synonymy, 6
synset, 4

thematic role grid, 22
token, 4
transition matrix, 53
transition probability, 53
type, 4

upper ontology, 8

valency frame, 22
variance, 43
variational inference, 48
verb valency, 22

weighted maximum bipartite matching problem, 135
word, 4
word sense disambiguation, 3
word sense induction, 3
word sketch, 146
word-form, 4