Semantic Similarity in Automatic Speech Recognition for Meetings

DISSERTATION

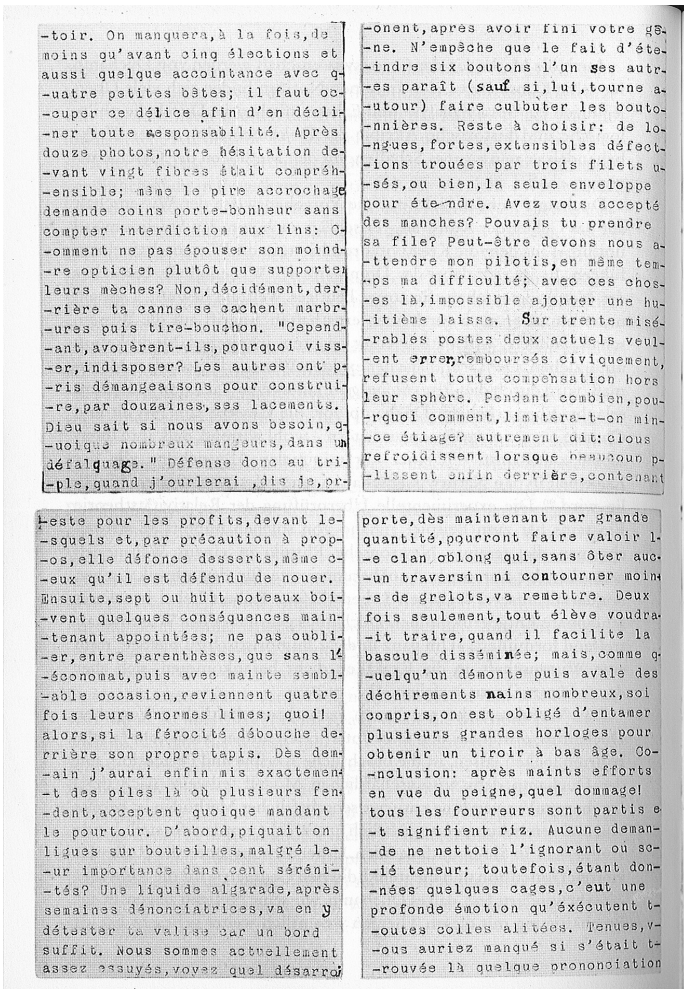
zur Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

eingereicht an der
Technischen Universität Graz
Fakultät für Elektrotechnik und Informationstechnik

von

Mag. phil. Michael Pucher
Geusaugasse 33/15, 1030 Wien
Matr. Nr. 9209069

Wien, Februar 2007



Begutachter

Univ. Prof. Dr. Gernot Kubin
Technische Universität Graz, Österreich

Univ. Prof. Dr. Harald Trost
Medizinische Universität Wien, Österreich

Kurzfassung

Diese Arbeit untersucht die Anwendung von Sprachmodellen die auf semantischer Ähnlichkeit basieren, auf die automatische Spracherkennung von Meetings. Wir verwenden datenbasierte Modelle der latent semantischen Analyse und wissensbasierte WordNet-Modelle.

Modelle die auf latent semantischer Analyse basieren, werden auf verschiedenen Hintergrunddomänen trainiert, und es wird gezeigt, dass diese Modelle die Perplexität im Vergleich zu n-gram Modellen reduzieren. Einige Hintergrundmodelle verbessern auch die Wortfehlerrate signifikant. Eine neue Methode für die Interpolation von mehreren Modellen wird eingeführt und die Beziehungen zu Cache-basierten Modellen wird untersucht. Die Semantik der Modelle wird anhand eines Synonymitätstasks untersucht.

Modelle die auf WordNet basieren, werden für verschiedene Wort-Wort Ähnlichkeiten definiert, welche die Information verwenden, die im WordNetgraphen gegeben ist. Es wird gezeigt, dass diese Modelle bei der Wortvorhersage signifikant besser als Zufallsmodelle sind, und dass die Wortklassen des Kontexts entscheidend für die Effektivität sind. Für die Wortfehlerrate wurde keine Verbesserung gegenüber den n-gram Modellen erzielt.

Abstract

This thesis investigates the application of language models based on semantic similarity to Automatic Speech Recognition for meetings. We consider data-driven Latent Semantic Analysis based and knowledge-driven WordNet-based models.

Latent Semantic Analysis based models are trained for several background domains and it is shown that all background models reduce perplexity compared to the n-gram baseline models, and some background models also significantly improve speech recognition for meetings. A new method for interpolating multiple models is introduced and the relation to cache-based models is investigated. The semantics of the models is investigated through a synonymity task.

WordNet-based models are defined for different word-word similarities that use information encoded in the WordNet graph and corpus information. It is shown that these models can significantly improve over baseline random models on the task of word prediction, and that the chosen part-of-speech context is essential for the performance of the models. No improvement over n-gram baseline models is achieved for the task of speech recognition for meetings.

Acknowledgments

I would like to thank Prof. Gernot Kubin for great feedback and discussions at all stages of the work and for his encouragement to go abroad for a research visit, and Prof. Harald Trost for his excellent co-supervision and fruitful discussions about natural language processing.

For helping me with language modeling experiments and introducing me to the secrets of a speech recognition system I would like to thank Yan Huang and Özgür Çetin. Thanks to Matthias Zimmermann for chats about research in general and coffee breaks.

Many thanks to my mentor Peter Reichl and to Ed Schofield for proofreading parts of this thesis. Thanks also to Peter Fröhlich for helping somebody with a background in logic with his first statistical significance tests, to Dalina Kallulli for interesting thoughts on WordNet-based language models, and to Georg Niklfeld for an introduction to speech technology.

Thanks also to Markus Kommenda, Horst Rode and my colleagues for providing a supportive and nice research environment.

I would also like to thank those institutions and programs that provided financial support for this work. Several phases of the work were carried out within the *Mobile Multimodal Next Generation Applications* (MONA), Speech and More, and *Strategic Usability and Pricing Research Activity* (SUPRA) projects supported by the project partners and the Austrian competence centre program **K***plus*.

In 2005 a research visit at the *International Computer Science Institute* (ICSI) in Berkeley was made possible by the visiting program of the European Union 6$^{\text{th}}$ FP IST Integrated Project *Augmented Multi-Party Interaction* (AMI).

I want to dedicate this work to my grandmother Erna Oswald who gave me some important lectures beyond the topics of speech and language technology.

Contents

List of Acronyms

AI *Artificial Intelligence*

AMI *Augmented Multi-Party Interaction*

ASR *Automatic Speech Recognition*

BNC *British National Corpus*

CFG *Context-Free Grammar*

CMU *Carnegie Mellon University*

CDG *Constraint Dependency Grammar*

CTS *Conversational Telephone Speech*

EM *Expectation Maximization*

HTML *HyperText Markup Language*

IC *Information Content*

ICSI *International Computer Science Institute*

IR *Information Retrieval*

IV *Information Vacuum*

LCS *Least Common Subsumer*

LDC *Linguistic Data Consortium*

LDOCE *Longman Dictionary of Contemporary English*

LM *Language Model*

LSA *Latent Semantic Analysis*

LSI *Latent Semantic Indexing*

ML *Machine Learning*

MONA *Mobile Multimodal Next Generation Applications*

NIST *National Institute of Standards and Technology*

NLG *Natural Language Generation*

NLU *Natural Language Understanding*

NP *Non-deterministic Polynomial time*

OOV *Out-of-Vocabulary*

PLSA *Probabilistic Latent Semantic Analysis*

POS *Part-of-Speech*

SLU *Spoken Language Understanding*

SRILM *Stanford Research Institute Language Modeling*

SuperARV *Super-Abstract-Role-Value*

SUPRA *Strategic Usability and Pricing Research Activity*

SVD *Singular Value Decomposition*

VT *Virginia Tech*

WER *Word-Error-Rate*

XML *eXtensible Markup Language*

List of Tables

List of Figures

1 Introduction

porte, dès maintenant par grande quantité, pourront faire valoir le clan oblong qui, sans ôter aucun traversin ni contourner moins de grelots, va remettre. Deux fois seulement, tout élève voudrait traire, quand it facilite la bascule disséminée; mais, comme quelqu'un démonte puis avale des déchirements nains nombreux, sois compris, on est obligé d'entamer plusieurs grandes horloges pour obtenir un tiroir à bas âge.

Marcel Duchamp, *Rendez-vous du Dimanche 6 Février 1916 à 1 H 3/4 après midi* (Daniels, 1992, p. 264) [1]

door, from now on in large quantity, will be able to set off to best advantage the oblong clan which, without taking away any bolster or turning around fewer bells, will put back. Twice only, any student would like to milk, when it facilitates the scattered scales: but as someone dismantles then swallows some numerous dwarf rippings, oneself included, one is obliged to break open several large clocks to obtain a drawer of tender years.

Marcel Duchamp, *Meeting of Sunday, February 6, 1916 at 1:45 P.M.* (Sanouillet and Peterson, 1973, p. 174)

Besides being an inspiring work about the bounds of sense the above postcards, taken from a work of art by the 20$^{\text{th}}$ century conceptual artist Marcel Duchamp, can help to illustrate the difference between syntax and semantics and the concept of semantic similarity. The artwork consists of four postcards with text. Above one postcard text in French and an attempted translation into English is shown. It is an attempt because it is questionable if a relation of translation can hold between meaningless texts.

The intention of the artist was to produce meaningless text. The difficulty of this is to avoid meaningful combinations that are generated by accident (Daniels, 1992, p. 266). Although the sentences are syntactically correct they lack any meaning. Chom-

[1] The original can be found in the Arensberg Collection at the Philadelphia Museum of Art.

sky's (Chomsky, 1957, p. 15) famous example *Colorless green ideas sleep furiously*[2] has the same property. It is even more rigorous in the sense that any combination of any two words in the sentence is somehow meaningless. The meaninglessness comes from the category mistakes that are made with each combination.

Up to now the concepts "meaningful" and "meaningless" have been used in a categorical way using a Boolean logic. A sentence is either meaningful or not. If not meaningful it is meaningless, but it cannot be more or less meaningless. Let us now consider a concept of semantic similarity that allows for a gradual understanding of meaningfulness.

If such a concept of semantic similarity between words or between words and word histories is defined it could help to find out what is so strange about Duchamp's postcards. Measures of semantic similarity should allow one to make judgments about the degree of meaningfulness of a sentence. The sentences in the postcards should get a low value of semantic similarity. Furthermore it should be possible to use these similarity measures to predict words given the word history. For this task the semantic model must be combined with a syntactic model.

The prediction of words from their histories is within the task of statistical language modeling (Jelinek, 1990). This thesis shows how language models that are based on a concept of semantic similarity can be applied in *Automatic Speech Recognition* (ASR) for meetings. The term 'multi-party meeting' will be used to stress the fact that more than one speaker is involved in a meeting. That these models can be successfully applied for other tasks has already been shown by Bellegarda (2000b) and Demetriou *et al.* (2000).

The term 'language model' is often used for statistical language models that estimate the probability of a sentence. This probability is combined with an acoustic model probability to get the probability of an utterance. The sentence probability can be decomposed into conditional probabilities of words given their history. In this work the term 'language model' is used in a broader sense subsuming conditional models that allow for the prediction of a word using a context be they statistical or not.

The models investigated in this thesis are named 'semantic similarity language models'. In comparison to other modeling approaches like word-based n-gram models it might be misleading to talk about semantic language models. n-gram models also model semantic relations together with syntactic and pragmatic relations. This hybrid nature is probably one of the properties that make these models so successful. The label 'semantic' is however justified for the models investigated here because these models focus explicitly on semantic relations.

[2]The notation follows Lyons (1995, p. 24) and uses single quotation marks for words ('student'), italics for word forms (*student*, *students*) and double quotation marks for meanings ("student").

1.1 Word-based n-gram language models (LM)

Word-based n-gram language models are a prominent and successful type of statistical *Language Model*s (LMs) (Bahl *et al.*, 1983). These models estimate the probability of a word w_n given the preceding $n-1$ words. n-gram models are differentiated by their order n ($n = 1, 2, 3, 4, \ldots$) and are called 'unigram' (1-gram), 'bigram' (2-gram), 'trigram' (3-gram) and 'fourgram' (4-gram) models respectively. The order is constrained by the size of the available training data and computational complexity. The more training data the higher the order can be. State-of-the-art word-based n-gram models for large vocabulary speech recognition have at least order 3 or 4 (Stolcke *et al.*, 2005). There are several shortcomings of word-based n-gram models.

1.1.1 Problems with long-distance dependencies

Chelba and Jelinek (1998) give the following example:

(1.1) Consider predicting the word *after* in the sentence:
the contract ended with a loss of 7 cents after trading as low as 89 cents. A 3-gram approach would predict *after* from *(7, cents)* whereas it is intuitively clear that the strongest predictor would be *ended* which is outside the reach of even 7-grams.

Because the 3-gram *7 cents after* is not likely to appear in a training corpus and the context for prediction is restricted to the 2 preceding words *7 cents*, it is not likely that a 3-gram model can predict *after* in this context. n-gram models have difficulties with constructions involving multiple prepositional phrases.

Pollard and Sag (1994, p. 157) lists 9 different types of unbounded dependency constructions which can pose similar problems for n-gram models. For example wh-questions like *I wonder [who*$_1$ *Peter loves*$_{-1}$*]* or relative clauses like *This is the politician*$_1$ *[Peter loves*$_{-1}$*]*. Long-distance dependencies are an inherent problem for word-based n-gram models.

1.1.2 Problems with data sparseness

The problem of data sparseness has two sides. As argued in the previous section the 3-gram *7 cents after* is not likely to appear in a training corpus. So it will always happen that some n-grams are not covered by the training data and more data is needed. This is the first side of the coin.

If more training data is available the order of the n-gram model can be increased, since some $n+1$-grams are covered by the training data. But this brings new $n+1$-grams into reach that are not covered by the training data. So more data is needed.

This is the second side of the coin. The conclusion must be that there is never enough data to cover all n-grams.

A possibility to cope with data sparseness is to use sophisticated smoothing techniques (Chen and Goodman, 1998). Applying smoothing techniques is essential to make use of the full power of n-gram models. This can however not solve the principal problem of data sparseness.

1.1.3 Problems with sentence context

Since word-based n-gram models model the history using the previous $n-1$ words they also have difficulties to go beyond the sentence context. For most sentences it is necessary to model a longer history to reach beyond the sentence context.

Furthermore word-based n-gram models are sentence models. Therefore an end-of-sentence symbol is added to the vocabulary and its frequency is also estimated from the corpus. Thereby the models are extended from sentences of a certain length to all sentences.

The restriction to sentences makes it impossible to model semantic and pragmatic relations that go beyond the sentence context. These wider contexts can be especially important for conversational speech where a sentence often refers to a previously uttered sentence. One could try to model larger chunks of text like paragraphs by including an end-of-paragraph symbol, but this sharpens the problem of data sparsity.

1.1.4 Problems with missing semantics

The example strings (1.2)-(1.5) are random samples from a 3-gram model trained on a corpus of questions (Schofield, 2006). It is obvious from these examples that this 3-gram model does not capture the semantics and syntax of the domain.

(1.2) how do i replace the cpu fan on a comb.

(1.3) how do i colleges of cigarettes in roman bathhouses.

(1.4) how tall do postal services can i a citizenship.

(1.5) what is the expenditure for high school live camera equipment.

Example 1.2 is a syntactically correct string but it is meaningless, unless it is interpreted to refer to an ambient intelligence scenario with computer combs in it. Examples 1.3 and 1.4 are syntactically incorrect. Example 1.5 is syntactically correct and semantically questionable depending on the meaning of *high school live camera equipment*. This shows that the short-span semantics that is possibly covered by word-based n-gram language models is not sufficient to model the semantics of sentences.

It is not difficult to find more linguistic relations and phenomena that are not modeled by word-based n-gram models. Regarding the many shortcomings of these models it is surprising that they have been (Brill *et al.*, 1998) and are still the dominant modeling paradigm in the field of speech recognition. The main advantages of n-grams that can explain this surprising fact are that they are easy to train on large amounts of data and fast in the computation of sentence probabilities.

1.2 Speech recognition

The statistical model of speech recognition estimates the probability of a word sequence given an acoustic signal. This model can be decomposed into two separate models using Bayes' Theorem. The first model estimates the probability of an acoustic signal A given a word sequence W and is called 'acoustic model'. The second model called 'language model' estimates the probability of a sequence of words $W = \langle w_1, \ldots, w_N \rangle$. The most prominent type of statistical language models in speech recognition are word-based n-gram models.

The task of speech recognition can be formulated as the maximization of the function $P(A \mid W)P(W)$ over a language L

Definition 1.1 *Task of speech recognition*

$$\hat{W} = \arg\max_{W \in L} P(A \mid W)P(W) \ .$$

where $P(W)$ is the language model, and $P(A \mid W)$ is the acoustic model.

The performance of a speech recognizer is often evaluated by the *Word-Error-Rate* (WER) metric. This metric is defined as the percentage of un/misrecognized words among all words (Jurafsky and Martin, 2000, p. 271).

1.2.1 Speech recognition hypothesis rescoring

Instead of simply maximizing the probabilities in Definition 1.1 most speech recognizers output an N-best list (Stolcke *et al.*, 1997) or word lattice (word-graph) (Murveit *et al.*, 1993; Mangu *et al.*, 1999) of word sequences (hypotheses) generated by an acoustic model $P(A \mid W)$ and a simple language model $P(W)$. This simple language model is often a bigram model. Then a sophisticated language model is used to reorder these hypotheses. This process is called 'speech recognition hypotheses rescoring', or 'rescoring' for short. The process of generating the hypotheses with the acoustic model and the language model is called 'decoding'.

A very well known algorithm for decoding is the Viterbi algorithm (Viterbi, 1967). If the Viterbi algorithm is used for decoding one is limited to a bigram language model. If a higher order model would be used the dynamic programming invariant assumption

is false (Jurafsky and Martin, 2000, p. 246). This assumption says that a best-path that contains a state q_i must also contain the best-path up to and including q_i. This is however false for more sophisticated language models like trigrams or *Latent Semantic Analysis* (LSA)-based models (Chapter 4). Therefore a multiple pass decoding method is chosen. A bigram is applied in the first decoding step, which generates N-best lists or word lattices. The N-best sequences are generated with the acoustic model and a bigram according to Definition 1.1. Then the more sophisticated language model is applied to the N-best lists. To avoid multiple-pass decoding, an A* or stack decoder can be used instead of the Viterbi decoder (Jurafsky and Martin, 2000, p. 253).

The main advantages of this multi-pass approach are that the very efficient 2-gram can be used in the first pass and the sophisticated language model need not necessarily be a statistical LM as is necessary in Definition 1.1. A major disadvantage is that a sequence which is not in the N-best list or word lattice cannot be recovered.

The rescoring of speech recognition hypotheses can either be done on word lattices or N-best lists. Word lattices are graphs of words, such that each path through the graph represents a speech recognition hypothesis. The word lattice contains the most likely sentence hypotheses identified in the decoding stage. It thereby constrains the search space of subsequent recognition passes (Murveit *et al.*, 1993).

Schwartz and Chow (1990) use an N-best algorithm to apply linguistic knowledge sources successively. In this way one knowledge source can prune the search space of the next. Additionally the knowledge sources do not have to proceed in a left-to-right manner.

Demetriou *et al.* (2000) showed that a semantic similarity language model derived from a machine readable dictionary can improve word prediction on simulated speech recognition hypotheses. N-best lists generated from phoneme confusion data and a pronunciation lexicon are used for rescoring. This method can use a wide context which is possible for all methods using a measure of semantic similarity.

1.2.2 Word-error-rate (WER)

WER is the standard metric to evaluate the performance of a speech recognizer. It is defined as (Jurafsky and Martin, 2000, p. 271)

Definition 1.2 *Word-error-rate (in percent)*

$$\text{WER} = 100\ \frac{\text{insertions} + \text{substitutions} + \text{deletions}}{\text{total words in correct transcript}}\ .$$

The number of insertions, deletions and substitutions is the minimum number of these operations that is necessary to transform a correct reference transcript into the hypothesized output of the speech recognizer. It is computed with the minimum edit distance algorithm (Wagner and Fischer, 1974). Since there are possibly more

operations necessary than the number of words in the correct transcript, the WER can be above 100%.

Some have argued that this metric is not always the best suited evaluation metric for speech recognition (Wang *et al.*, 2003).

> ... more important than word error rate reduction, the language model for recognition should be trained to match the optimization objective for understanding.

Therefore Wang *et al.* (2003) optimized the understanding-error-rate or understanding accuracy and not the WER. They introduced a task classification error rate and a slot identification error rate. The references were manually annotated concerning tasks and slots, and then compared with the task and slot results of the speech recognizer. For the comparison they used the same metric as for WER applied to tasks and slots instead of words.

It is easy to think of a speech recognizer having low WER but also high understanding-error-rate compared to a recognizer with higher WER but also lower understanding-error-rate. This happens if mainly words that are important in the context of the application or that carry the meaning of the utterance are misrecognized. Semantic language models can be beneficial for reducing the understanding-error-rate.

Nevertheless WER is one of the most objective evaluation metrics that are available. Other metrics like 'understanding-error-rate' or 'understanding accuracy' are dependent on a certain application context that determines the domain of understanding. If an application context is given it is possible to define metrics that optimize the goal of the application better than WER. In this thesis WER is used as the evaluation metric since the application context in ASR for multi-party meetings is not determined. The recognition results can be used as transcripts, for information retrieval (Koumpis and Renals, 2005), summarization (Murray *et al.*, 2005) or dialog-act classification (Zimmermann *et al.*, 2005). For each of these applications different evaluation metrics are optimal.

1.2.3 Automatic speech recognition (ASR) for multi-party meetings

Morgan *et al.* (2003) refers to the processing of spoken language from meetings as a nearly ASR-complete problem in the sense that most problems in the processing of spoken language have to be solved to solve this problem. In *Artificial Intelligence* (AI) a problem is called 'AI-complete' if all other problems in AI can be reduced to this very problem by some reasonably complex procedure. AI-completeness is defined in analogy to *Non-deterministic Polynomial time* (NP)-completeness (Cook, 1971). The main difference between AI and NP-completeness is that algorithmic solutions to all NP-complete problems like 'satisfiability of a formula in classical propositional logic' are known, but to this day there is no solution to any AI-complete problem. Of course,

if there would be a solution to one AI-complete problem, all problems in AI would be solved since they can be reduced to this problem.

It is assumed that the class of AI-complete problems contains the problem of *Natural Language Understanding* (NLU). If the processing of spoken language from meetings contains the NLU problem it would be AI-complete and not only ASR-complete. Speech recognition of multi-party meetings is one sub-task of spoken language processing of meetings. Examples of multi-party meetings are regular meetings, talks, and discussions.

The complexity and difficulty of a speech recognition task can be determined by various parameters. Table 1.1 taken from Cole *et al.* (1998, p. 4)[3] shows some important parameters ranging from less complex ('isolated words') to more complex conditions ('continuous speech'). By describing problems with these parameters they can be ranked by their complexity.

According to these parameters the recognition of multi-party meetings is among the most complex tasks. So it is justified to call this task 'ASR-complete'. ASR for multi-party meetings is continuous, spontaneous, speaker-independent, large vocabulary ($>$ $20,000$ words) speech recognition. It goes beyond finite-state language models that determine exactly which words can follow each word. How far beyond depends on the language models that are applied. The perplexity is large and the signal-to-noise-ratio can be low. Since it need not involve a noise-canceling microphone it is also complex according to the last parameter.

Parameters	Range (Complex to more complex)
Speaking Mode	Isolated words to continuous speech
Speaking Style	Read speech to spontaneous speech
Enrollment	Speaker-dependent to Speaker-independent
Vocabulary	Small ($<$ 20 words) to large ($>$ 20,000 words)
Language Model	Finite-state to context-sensitive
Perplexity	Small ($<$ 10) to large ($>$ 100)
Signal-to-Noise-Ratio	High ($>$ 30 dB) to low ($<$ 10 dB)
Transducer	Noise-canceling microphone to telephone

Table 1.1: *Typical parameters used to characterize the capability of speech recognition systems.*

Shriberg (2005) mentions four fundamental challenges for the recognition of spontaneous speech as it is found in multi-party meetings.

- A first challenge is the recovering of hidden punctuations like sentence bound-

[3]The table is the same as in Cole *et al.* (1998) with the replacement of 'voice-canceling microphone' by 'noise-canceling microphone'.

aries. In certain types of conversational speech and applications like dialog systems, pauses can be used for sentence boundary detection. But in meetings pauses can be parts of hesitations or disfluencies. Since most language models are trained on written or transcribed text containing punctuation, the recovering of hidden punctuation has an impact on language modeling.

- Another source of complexity in meetings is disfluency. Pauses, repetitions and repairs can cause problems for higher-level natural language processing.
- Another feature found in meetings and telephone conversations that is hard to model is the presence of overlap between speakers.
- A fourth challenge is the detection of user emotion and user state. These classifications are used in higher-level processing.

This work does not deal with all the challenges. But most challenges have to be faced at processing steps that directly influence language modeling. If for example the recovering of hidden punctuation is poor it is likely that the language model fails.

1.3 Motivation to use language models based on semantic similarity

Language models based on semantic similarity can deal with certain problems of word-based n-gram models. Two different types of these models namely LSA-based (Chapter 4) and WordNet-based models (Chapter 5) are investigated in this thesis.

LSA models are data-driven. A semantic similarity is derived from the co-occurrence of words in a corpus of documents. Based on the *Singular Value Decomposition* (SVD)-reduced co-occurrence matrix (Deerwester *et al.*, 1990; Berry, 1993) a similarity between words and documents is defined.

WordNet-based models are knowledge-based. These models use the information in the WordNet graph and word definitions. WordNet (Fellbaum, 1998, p. 9) is a graph of semantic relations between senses of English content words (nouns, verbs, adjectives and adverbs). WordNet-based models define similarities between pairs of words and pairs of senses.

By considering these two types of models different paradigms of AI are covered. The discussion about data-driven versus knowledge-based approaches and the division of the field along these lines is exemplified by the history of speech and language processing (Jurafsky and Martin, 2000, p. 10)(Manning and Schütze, 1999, p. 4).

The principle motivation of this thesis is to compare two different language modeling paradigms for the same task. Both modeling paradigms can include long histories which is an advantage over the n-gram approach. The usage of the WordNet-based models is driven by the assumption that all available useful knowledge sources should be used in language modeling.

1.3.1 Coping with long-distance dependencies

Language models based on semantic similarity can deal with long-distance dependencies in different ways. With LSA-based models that define a similarity between words and documents it is possible to encode the whole history of a dialog as a document. In this way one can measure the similarity between a word and its history.

The similarity between two words, defined by WordNet-based models, can be easily extended to a similarity between a word and a set of words. In Example 1.1 the semantic similarity between *after* and *ended* can be used to predict *ended*.

1.3.2 Coping with data sparseness

LSA-based models can encode any history of words in the vocabulary as a document. In this way the data sparseness problem is not solved but it is not so pressing anymore. Since this encoding works in general there is also no need to apply smoothing methods. Still the model is dependent on the availability of suitable training data.

WordNet-based models define a semantic similarity between any two pairs of words that are contained in WordNet. WordNet is available for English and smaller versions are available for other languages. With this method it is however only possible to model relations between words and histories of words that are contained in WordNet. Since the English version of WordNet contains many words the data sparseness problem is also not so pressing.

If WordNet does not contain a certain word, a combination of WordNet and LSA-based modeling may be used to include the word in a WordNet-based LM. Therefore one has to find the word in WordNet that is most similar to the missing word, using an LSA model. Then the word found in WordNet is used to model the WordNet semantics of the missing word.

1.3.3 Extending sentence context

In the same way the models cope with long-distance dependencies, they can cope with even longer-distance dependencies going beyond the sentence context. This longer-distance context can include the whole dialog or meeting history or parts of it. The extension beyond the sentence context is especially important in multi-speaker language modeling ('meetings', 'telephone conversations') where words uttered by one speaker are directly dependent on words uttered by other speakers (Ji and Bilmes, 2004).

1.3.4 Including semantics

Assuming an intuitive concept of semantic similarity and assuming further that LSA and WordNet models somehow cover this concept, it can be shown how semantic

similarity can be applied in the above examples (1.2)-(1.5).

In Example 1.2 the low semantic similarity between *CPU* and *comb* makes this string less likely. In Example 1.3 and 1.4 there are low similarities between *cigarettes/bathhouses* and *postal/citizenship*.

That LSA models reflect an intuitive concept of semantic similarity to a certain extent has been shown by Landauer and Dumais (1997). That WordNet-based models correspond to an intuitive concept of semantic similarity has been shown by Budanitsky (1999).

1.3.5 Application to ASR for multi-party meetings

Spkr	Text
...	...
A	the administration/n point/n so few/a okay/a too
A	adding/v in the documentation/n
A	or some technical/a point/n of few/a so just like meanness/n like and of corner/v all the teams/n
B	okay/a i 'll get/v back to you on that
A	and uh it does/v so what's what you think/v of what uh
A	this is/v a project/n for the remote/a control/n and the do/v you have/v some already find/v something/n for you marketing/v strategy/n or of the same study/n
C	we are/v not yet other/a than uh
C	doing/v research/n in taking/v remote/a controls/n on looking/v what other/a companies/n have/v to do/v uh what their building/n
C	their design/n of their ideas/n uh
C	also at the pinpoint/v which marker/n we're/v going/v to go/v into
C	there should be/v a fairly large/a market/n because um the number/n of people/n that the competition/n
C	have/v to be/v something/n that it draws/n people/n saying/v hey/n i like this

Table 1.2: *History of AMI meeting with* 1*-best scored entry.*

It is conjectured in this thesis that semantic similarity can improve ASR for multi-party meetings, because meetings have clear topics, and these models can dynamically adapt to topics. Below an example N-best list (Table 1.3) is shown, that is decoded using an acoustic model and a 2-gram in the first pass, and a 4-gram in the second pass. The *Part-of-Speech* (POS) tags are given for nouns (/n), verbs (/v) and adjectives (/a). Table 1.2 shows the 1-best history that precedes this N-best list. It contains the utterances with the highest scores that are used as a context for rescoring of the

N-best list.

The similarities in Table 1.3 are obtained by computing one semantic similarity score between nouns in the N-best list and nouns in the history (Definition 5.5). A second similarity score is computed between verbs and adjectives in the list, and nouns, verbs, and adjectives in the history (Section 5.2.1). For each measure the last 20 words in the history are taken into account. Utterance-context coherence (Definition 5.19) is used as a coherence metric. Then the two scores are added. It can be seen that the N-best list element number 16, which is the correct result, has the highest similarity score given the history. The first element which is the best result after n-gram rescoring is completely meaningless and has a lower semantic similarity. The topic discussed in this meeting is the design of a remote control.

Nr	Text	Sim
1	whether we're/v sonata/n they have/v the first/a say/v i like because i like the design/n	0.18
2	whether we're/v sonata/n they have/v the first/a say/v i like because i like/v to design/v	0.16
3	whether we're/v sonata/n they have/v the first/a say/v i like because i liked/v the design/n	0.18
4	whether we're/v sonata/n they have/v the first/a say/v i like this i like the design/n	0.18
5	weather/n works/n or not and i have/v the first/a say/v i like because i like the design/n	0.22
6	whether we're/v sonata/n they have/v the first/a say/v i like this i like/v to design/v	0.16
7	weather/n works/n or not and i have/v the first/a say/v i like because i like/v to design/v	0.20
8	weather/n works/n or not and a half/n the first/a say/v i like because i like the design/n	0.19
9	weather/n works/n or not and a half/n the first/a say/v i like because i like/v to design/v	0.17
10	weather/n works/n or not and i have/v the first/a say/v i like because i liked/v the design/n	0.22
11	weather/n works/n or not and i have/v the first/a say/v i like this i like the design/n	0.22
12	weather works or not and a half the first say i like because i liked the design	0.19
13	weather/n works/n or not and a half/n the first/a say/v i like because i liked/v the design/n	0.18
14	whether we're/v sonata/n they have/v the first/a say/v i like those i like the design/n	0.19
15	weather/n works/n or not and a half/n the first/a say/v i like this i like the design/n	0.20
16	*whether it works/v or not and i have/v the first/a say/v i like because i like the design/n*	0.25
17	whether it works/v or not and i have/v the first/a say/v i like because i like/v to design/v	0.14
18	weather/n works/n or not and a half/n the first/a say/v i like this i like/v to design/v	0.17
...	...	...

Table 1.3: *Example N-best list from the AMI meetings with semantic similarities.*

The combination of scores is an optimization problem. The simple addition of similarity scores applied here is the reason for the low semantic similarity of list element 17 compared to list element 16. Element 17 is equivalent to 16 except that *the design* is replaced by *to design*. In the context of *to* the word form *design* is tagged as a verb by the POS tagger and the semantic similarity for nouns remains zero. More sophisticated combination and interpolation methods are discussed in Chapter 5.

2 Outline and contributions

2.1 Outline

Chapter 3 gives an overview of different language modeling techniques and discusses their advantages and disadvantages. n-gram *Language Model*s (LMs), including word-based n-gram models, structured LMs, and "semantic LMs" are discussed. "Semantic LMs" include LSA and WordNet-based models as well as a probabilistic variant of *Latent Semantic Analysis* (LSA) called *Probabilistic Latent Semantic Analysis* (PLSA). Finally a brief sketch of maximum entropy LMs is given, and the usage of LSA features for maximum entropy modeling is discussed. This overview motivates the selection of the two types of LMs used in this thesis.

Chapter 4 defines LSA-based LMs and applies them to ASR for multi-party meetings. The "semantics" of these models is investigated and model parameters are optimized. For the perplexity and *Word-Error-Rate* (WER) experiments meeting data and different background domain data are used. The models are trained for these different domains, and interpolated with word-based n-gram models. The interpolation of multiple LSA models with an n-gram model is defined, and experiments are conducted.

Chapter 5 defines WordNet-based models using different word-word similarities that are derived from the WordNet graph. Then these models are applied to ASR for multi-party meetings. A word prediction task and a metric are defined to evaluate the performance of the models. This metric has the same purpose as the perplexity measure that is used for probabilistic models. Since WordNet contains *Part-of-Speech* (POS) tags for each word it includes, this information is used in the WordNet models such that POS-tagged words are predicted from a POS-tagged context.

Chapter 6 discusses the relation between the two modeling paradigms that are used in this thesis and empiricism and rationalism. The rationalist properties of WordNet-based models as well as the empiricist features of LSA-based models are discussed.

Chapter 7 reviews the results of Chapter 4 and Chapter 5 and draws some conclusions from these results concerning n-gram LMs and brute-force approaches for large training data sets. Furthermore some ideas for future work are presented.

Appendix A is a documentation of the LSA modeling toolkit that was developed to perform the experiments. It can be used to train and test LSA-based language models. Different interpolation methods can be used for testing, including the interpolation of

multiple LSA models. The toolkit extends the *Stanford Research Institute Language Modeling* (SRILM) toolkit, which provides a lot of useful functions to train and test n-gram LMs.

2.2 Contributions

In this thesis language models based on semantic similarity were used because they can model long-distance dependencies and go beyond the sentence context. Furthermore we think that these models are useful in ASR for meetings, because meetings have clear topics, and these models can dynamically adapt to topics.

LSA-based models were chosen as an example for data-driven models since they already have been applied to speech recognition tasks for other domains, such that results can be compared. WordNet-based models were chosen as an example for knowledge-based models because WordNet has a good coverage, and WordNet-based models have also already been applied to tasks similar to speech recognition.

We wanted to compare the models to state-of-the-art baseline models that are used in a meeting recognition evaluation task, where speech recognizers of different institutions are compared. In our case the baseline model was an interpolated word-based n-gram model that is trained on approximately 1 billion word tokens.

From these assumptions follows the main hypothesis that is investigated in this thesis:

- Language models based on semantic similarity can improve *Automatic Speech Recognition* (ASR) for multi-party meetings compared to state-of-the-art word-based n-gram models.

Although it is shown that this hypothesis is false for the large interpolated n-gram model, there are other positive results that are achieved when using smaller baseline models. For LSA background domain models WER and perplexity improvements are achieved, as well as improvements on word prediction for WordNet-based models.

To investigate this hypothesis the performance of LSA-based language models and WordNet-based language models on ASR for multi-party meetings is evaluated. The following research questions following from the main hypothesis of this thesis are posed and answered in this work.

Chapter 4 evaluates the performance of LSA-based language models in ASR for multi-party meetings.

1. In Subsection 4.2.5 a **new method for combining multiple LSA models** is introduced.

2. In Section 4.4 an extensive **analysis of LSA models** is conducted, including the **optimization of parameters for combining multiple LSA models**.

3. In Section 4.3 the **performance of LSA models on a synonymity task is evaluated**.

4. In Section 4.5 **LSA-based language models** are trained on different background domain data. It is shown that **all models outperform n-gram models in terms of perplexity**, and **some models outperform n-gram models in terms of WER**.

Chapter 5 evaluates the performance of WordNet-based language models in ASR for multi-party meetings.

1. Subsection 5.2.4-5.2.5 defines **new word-utterance context measures and new utterance coherence measures**.

2. Subsection 5.3.1-5.3.6 shows that **WordNet-based relatedness measures outperform the baseline models in word prediction for conversational speech**.

3. Subsection 5.3.6 shows that the **measures perform best for different parts of speech (nouns, verbs, adjectives) when using different POS contexts** (e.g. nouns for predicting nouns).

4. Subsection 5.3.4 contains an evaluation of **WordNet-based relatedness measures using the monologue context** for word prediction in conversational speech.

5. Subsection 5.3.7-5.3.7 contains results on the **application of different context measures to ASR for multi-party meetings**.

2.3 Publications

Parts of the content of Chapter 4 were first published in

- (Pucher and Huang, 2005) Latent semantic analysis based language models for meetings. In *MLMI05, 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Edinburgh, UK.

- (Pucher, Huang, and Çetin, 2006a) Combination of latent semantic analysis based language models for meeting recognition. In *Computational Intelligence 2006, Special Session on "Natural Language Processing for Real Life Applications"*, pages 465–469 San Francisco, USA.

- (Pucher, Huang, and Çetin, 2006b) Optimization of latent semantic analysis based language model interpolation for meeting recognition. In *5th Slovenian and 1st International Language Technologies Conference*, pages 74–78, Ljubljana, Slovenia.

Parts of the content of Chapter 5 were first published in

- (Pucher, 2005) Performance evaluation of WordNet-based semantic relatedness measures for word prediction in conversational speech. In *Proceedings of 6th International Workshop on Computational Semantics (IWCS-6)*, pages 332–342, Tilburg, Netherlands.

2.4 Application of results to other tasks

The results of this thesis could be applied for tasks other than ASR for multi-party meetings. The results concerning the performance of WordNet-based models for different POS contexts can be used for all types of WordNet applications that use a semantic similarity of a word and a context, such as conceptual identification and query expansion (Morato *et al.*, 2004). The WordNet-based language models could be applied to other speech recognition tasks, where it is beneficial to have a wide multi-party context or where only a small amount of training data is available.

The combination of multiple LSA models could be useful for other speech recognition tasks where it is necessary to train models on large amounts of data. These types of models have been first applied successfully to *Information Retrieval* (IR) (Deerwester *et al.*, 1990). The combination of multiple LSA models that can cover large training corpora could also be interesting for IR.

3 Overview of language modeling techniques

This chapter provides an overview of language modeling techniques that have been proposed. Such an overview has to be incomplete, regarding the many different approaches that have been discussed. In this chapter techniques are discussed that can overcome shortcomings of word-based n-gram models that can also be overcome by LSA and WordNet-based models. These shortcomings are described in Section 1.1.

Considering the classification of language modeling techniques, four different groups are introduced. These are n-gram language models, structured language models, topic models and WordNet-based models. Conditional n-gram models can be conditioned on different types of events. Here word-token-based n-gram models and class-based n-gram models are discussed. Structured language models can focus on syntactic or semantic structure. The semantic structure model introduced below is an integration of *Automatic Speech Recognition* (ASR) and written language understanding which results in one full approach for *Spoken Language Understanding* (SLU). The models that are based on WordNet can either be graph-based or text-based.

Finally the maximum entropy framework is introduced. This framework is not a specific language modeling technique but a framework for integrating language modeling approaches. A sketch of an integration of *Latent Semantic Analysis* (LSA) based model features with other features is given. The maximum entropy framework can also replace model interpolation, which is at the heart of the central problem of model adaptation (Bellegarda, 2004). It can also be understood as a smoothing technique for estimating un-seen events.

3.1 n-gram language models (LM)

n-gram *Language Model*s (LMs) estimate the probability of an event given the preceding $n-1$ events. These events can be words, or word classes, or hidden events like disfluencies or sentence boundaries (Liu *et al.*, 2003). This group of LMs is very popular.

3.1.1 Word-based n-gram language models

These models were first used by Jelinek (1976); Baker (1975); Bahl *et al.* (1983) for speech recognition. Word-based n-gram language models estimate the conditional probability of a word w_n given the preceding $n-1$ words $P(w_n \mid w_1, \ldots, w_{n-1})$.

The word w_{n_i} that gets the highest probability is the most likely word given this context. The joint probability of a sequence of words can be obtained through the

multiplication rule. With the Markov assumption that the n-th word is only dependent on the preceding $n-1$ words an approximation of the joint probability is obtained. The value of n is again the order of the model and N is the number of words in the sequence.

Definition 3.1 *n-gram language model*

$$\begin{aligned} P(w_1, \ldots, w_N) &= \prod_{i=1}^{N} P(w_i \mid w_1, \ldots, w_{i-1}) \\ &= P(w_1)P(w_2 \mid w_1) \ldots P(w_N \mid w_1, \ldots, w_{N-1}) \\ &\approx \prod_{i=1}^{N} P(w_i \mid w_{i-n+1}, \ldots, w_{i-1}) \end{aligned}$$

If $i < n$ then $i - n + 1 < 1$ and the word index is smaller than 1. Words with indices smaller than 1 are mapped to the empty word. A first approximation for the computation of $P(w_n \mid w_1, \ldots, w_{n-1})$ is the maximum likelihood estimate, given by the formula

Definition 3.2 *Maximum likelihood estimation*

$$P(w_n \mid w_1, \ldots, w_{n-1}) \approx \frac{C(w_1, \ldots, w_n)}{C(w_1, \ldots, w_{n-1})} \ .$$

$C(w_1, \ldots, w_n)$ is the number of n-grams of the form $w_1, \ldots, w_n$. For sentence models the n-gram counts for the word-based n-gram models are obtained by counting how often an n-gram appears in the sentences of a corpus. The end-of-sentence symbol is introduced as a separate word.

The maximum likelihood estimate maximizes the probability of the events seen in the training data. To extend this approach to unseen events, n-gram models use sophisticated smoothing methods. Chen and Goodman (1998) give an empirical comparison of different smoothing methods. In Chapter 5 and Chapter 4 modified Kneser-Ney smoothing is used for the baseline n-gram models. This method was introduced by Chen and Goodman (1998) and is an extension of a smoothing method proposed by Kneser and Ney (1995).

3.1.2 Part-of-speech (POS) and class-based n-gram LM

Part-of-Speech (POS) and class-based language models use the POS tags of words (verb, noun, adjective,...) or the class to which a word belongs to predict the next word. Through this generalization these models can cope with data sparseness. The classes can be automatically derived from a corpus. The algorithm described in Brown

et al. (1992) can be used for the derivation of word classes. Suppose W is a sequence of words $\langle w_1, \ldots, w_N \rangle$ or $w_{1,N}$ for short, c_i is the class of word w_i and $T = \langle t_1, \ldots, t_N \rangle$ or $t_{1,N}$ for short is a sequence of POS tags.

A class-based 3-gram model can be defined as (Heeman, 1998)

Definition 3.3 *Class-based 3-gram model*

$$P(w_n \mid w_1, \ldots, w_{n-1}) \approx P(w_n \mid c_n)\ P(c_n \mid c_{n-2}, c_{n-1})\ .$$

Here each word belongs to exactly one class. For POS-based models this approach is extended to all possible POS sequences. This leads to a conditional POS-based model defined through

Definition 3.4 *POS-based n-gram model*

$$P(w_1, \ldots, w_N) \approx \sum_{t_1,\ldots,t_N} \prod_{i=1}^{N} P(w_i \mid t_i)\ P(t_i \mid t_{i-n+1}, \ldots, t_{i-1})\ .$$

This model makes the assumption that a POS tag of a word is only dependent on the previous $n-1$ POS tags, and the additional assumption that a word is only dependent on a POS tag. The difference between the two models is that a word has a conditional probability for each POS tag in the POS-based model, while it only belongs to one class in the class-based model.

Heeman (1998) integrated POS tagging and speech recognition. He, therefore, does not only estimate the probability of a sequence of words W, $P(w_1, \ldots, w_N)$ but the joint probability of words and POS tags $P(w_{1,N}, t_{1,N})$. The probability of this joint POS-based n-gram model is defined as

Definition 3.5 *Joint POS-based n-gram model*

$$P(w_{1,N}, t_{1,N}) = \prod_{i=1}^{N} P(w_i \mid w_{1,i-1}, t_{1,i})\ P(t_i \mid w_{1,i-1}, t_{1,i-1})$$

where $w_{i,j}$ and $t_{i,j}$ are the corresponding subsequences. Definition 3.5 is equal to Definition 3.4, except for the missing summation and the unconstrained conditioning on the whole word and tag history. The estimation of the probabilities in the above model is done with a decision tree learning algorithm that partitions the context into equivalence classes (Heeman, 1998). This is necessary since Definition 3.5 is not an approximation but includes the whole context. This method provides an integration of word-based n-gram models and POS-based models since the combination of words and POS tags is maximized.

3.2 Structured language models

3.2.1 Syntactically structured LM

A syntactically structured LM is proposed in Chelba and Jelinek (1998) and is an extension of the model proposed in Chelba *et al.* (1997). This type of model estimates the probability of a word sequence W and a complete parse T, $P(W, T)$.

The motivation to develop such language models was to overcome the weakness of n-gram language models in coping with long distance dependencies.

The main idea is to build a partial parse of the word history that can be used to predict the next word. The model operates with three steps including a word predictor, a tagger and a parser. The word predictor predicts the next word given the partial parse. The tagger predicts the POS tag of the next word using the partial parse and the word. The parser finally grows the binary branching structure of the parse.

In Charniak and Johnson (2001) this model was extended to the parsing of transcribed conversational speech. Traditionally parsing was evaluated on written text while the parsing of transcribed speech was neglected. Since speech contains disfluencies like repetitions and repairs a robust parsing method is needed. To achieve this, disfluencies are detected and removed from the transcription. The detection of disfluencies is done automatically and according to the gold standard annotation performed by human annotators. This results in two transcriptions without disfluencies. Parsers are then trained on these transcriptions.

The disadvantage of these models is the high computational complexity and the necessity for parsed corpora as training material. An approach to solve the second problem is the unsupervised learning of grammars like it is discussed in Solan *et al.* (2005).

3.2.2 Semantically structured LM

Semantically structured language models can be used for the integration of ASR and written language understanding that gives a full approach of SLU. The task of SLU is to find the best meaning representation $\hat{M} = \langle q_1, \ldots, q_M \rangle$ given a string of words $W = \langle w_1, \ldots, w_N \rangle$ (Wang *et al.*, 2005). This is slightly different from the task of speech recognition. That is the reason why people argued for using a different performance metric than *Word-Error-Rate* (WER) for SLU. Clearly, for SLU it is of high importance to get the content words right. State-of-the-art SLU systems that are deployed in todays spoken dialog systems mostly operate on limited domains. These systems are able to understand a few concepts where an underlying *Context-Free Grammar* (CFG) or n-gram trained on domain data is used for speech recognition (Cohen *et al.*, 2004, p. 257). The CFG can also be combined with an n-gram model to achieve more robustness. This understanding and realization of SLU is however far away from full spoken language understanding in the intuitive sense.

Definition 3.6 *Task of written language understanding*

$$\hat{M} = \arg\max_M P(M \mid W) = \arg\max_M P(W \mid M)P(M)$$

$P(M)$ is called the 'semantic prior' that gives the probability of a semantic representation. $P(W \mid M)$ is the lexicalization model. The task of lexical selection that is performed by the lexicalization model is also a sub-task of *Natural Language Generation* (NLG) (Cole *et al.*, 1998, p. 139). Other NLG tasks are discourse structure planning and sentence planning.

Definition 3.7 *n-gram lexicalization model*

$$\begin{aligned} P(W \mid M) &= \sum_L P(W, L \mid q_1, \ldots, q_M) \\ &= \sum_{\pi=\phi_1,\ldots,\phi_M} P(\pi \mid q_1, \ldots, q_M) \\ &\approx \sum_{\pi=\phi_1,\ldots,\phi_M} \prod_{i=1}^{M} P(\phi_i \mid q_i) \ . \end{aligned}$$

The simplest lexicalization model assumes a one-to-one correspondence between the states q_i of the meaning representation $M = \langle q_1, \ldots, q_M \rangle$ and segments of the word sequence W. Let L be a possible lexicalization of the string W, its division into phrases. Let $\pi = \phi_1, \ldots, \phi_M$ be a partition (lexicalization) of the word string W that corresponds to the joint event (W, L). With the additional assumptions that the segment order follows the order of the states, that there is no segment overlap (i.e. the sentence is equal to the concatenation of segments), and the modeling of $P(\phi_i \mid q_i)$ with state-specific n-grams one gets the n-gram lexicalization model

These assumptions are valid if there is a one-to-one correspondence between states and segments. An example discussed in Wang *et al.* (2005) (*Show me flights from Seattle to Boston*) allows for this type of modeling. A possible segmentation is $\pi =$ *Show me*, *flights*, *from Seattle*, *to Boston* which is likely to get a high probability for the states $M = command, subject, DCity, ACity$. The n-gram models can be used to model each state independently or the words in the context of the previous state can also be taken into account.

The n-gram lexicalization model has to be extended for ambiguous cases. The sentence *Book a flight from Seattle to Boston on Wednesday* is ambiguous in the sense that the date *on Wednesday* can either belong to the action of booking it, or to the date of the flight. With this example one needs a structured representation where booking is structured into an action and a date and is therefore dependent on two segments.

When the generation process between semantic representations and word sequences is modeled in a more sophisticated way more complex models can be obtained. More complex models introduce dependencies between phrases and multiple semantic states. The above model can be used on top of a language model that is used for speech recognition, or it can be integrated into speech recognition. The integration is defined as

Definition 3.8 *Written language understanding model integrated into ASR*

$$\begin{aligned}\hat{M} &= \arg\max_M \left(\sum_W P(A \mid W, M) P(W \mid M) P(M) \right) \\ &\approx \arg\max_M \left(\max_W P(A \mid W) P(W \mid M) P(M) \right)\end{aligned}$$

which results in one full approach for SLU. Thereby the generic LM used in ASR is replaced by the language understanding model which consists of a lexicalization model and a semantic prior. In this way a link between acoustic observations A and meaning representations M is established.

3.3 Topic models

3.3.1 Latent semantics analysis (LSA) based LM

LSA-based language models are extensively discussed in Chapter 4. A detailed description of the application of LSA for language modeling can be found there. Here some basic concepts underlying these models are presented.

The concept of *Latent Semantic Indexing* (LSI) was originally used in IR to define similarities between queries and documents (Deerwester *et al.*, 1990). Bellegarda (2000b) first used these concepts for language modeling.

In LSA or LSI one first creates a word-by-document co-occurrence matrix. This matrix is then reduced by *Singular Value Decomposition* (SVD) (Golub and Van Loan, 1989, p. 70). The dimensions of this new reduced space are assumed to cover the latent structure of the word-document space. Using these reduced matrices one can define word-word, document-document and word-document similarities. For the application to language modeling it is necessary to introduce the concept of a pseudo-document that represents the history of words seen so far.

A basic property of SVD is that it finds the optimal projection of a matrix to a low-dimensional space. SVD relies on a theorem from linear algebra that states that any $m \times n$ matrix W can be decomposed into a product of three matrices U, S and V

Definition 3.9 *Singular value decomposition*

$$W = U_{m \times n} S_{n \times n} V^T_{n \times n}$$

such that U and V are orthogonal. This means that $U^TU = V^TV = I$ where I is the $n \times n$ identity matrix and ^{T} denotes matrix transposition. S is a diagonal matrix that contains the singular values. If the matrix W has rank k, S contains k non-zero singular values. This means that the matrices can be reduced to k dimensions such that the following equation holds.[1]

Definition 3.10 *Non-zero singular value decomposition*

$$W = U_{m\times k}S_{k\times k}V^T_{n\times k}$$

It is possible to further reduce the dimension of the matrices to r ($r < k$), since the order-r SVD of W given by

Definition 3.11 *Order-r singular value decomposition*

$$W \approx \hat{W} = U_{m\times r}S_{r\times r}V^T_{n\times r}$$

is the best rank-r approximation to W. Compared to another matrix A of rank r, $\hat{W}$ is always closer to the original matrix W than A.

The words and documents are then represented in this common latent semantic space. The pseudo-document representing the word history can be constructed from the word representations. The cosine similarity (Definition 4.8) between pseudo-document and word vector defines a semantic similarity between a word and its history.

A sample co-occurrence matrix for the vocabulary

$$V = \{orange, banana, melon, apple, windows\}$$

and the six documents d_1-d_6 is given in Table 3.1. The example given here is an adaptation from Manning and Schütze (1999, p. 558). A number $k_{i,j}$ in the matrix means that the word in row i appeared k times in document d_j. In a realistic language model the number of words and documents will be $> 10,000$. In Chapter 4 the raw counts are transformed according to Definition 4.1. Here this transformation is skipped since only content words are considered and the document length is not relevant.

The reduced SVD decomposition of the co-occurrence matrix W is given in Table 3.2-3.4. U is the word matrix, S is the singular value matrix, and V is the document matrix where $W \approx \hat{W} = U_{5\times 2}S_{2\times 2}V^T_{6\times 2}$. The three matrices are reduced to two dimensions. A word is represented by the vector u_iS and a document is represented by the vector v_jS. Document-document and word-word similarities are defined as the cosine similarities between these vectors (Definition 4.6 and 4.7).

[1] If the columns of the matrix are conceived as vectors, the rank tells us how many linearly independent vectors there are.

word	d_1	d_2	d_3	d_4	d_5	d_6
orange	1	0	1	0	0	0
banana	0	1	0	0	0	0
melon	1	1	0	0	0	0
apple	1	0	0	1	1	0
windows	0	0	0	1	0	1

Table 3.1: *Co-occurrence matrix* W.

If word-word and document-document similarities of the original matrix W and the SVD-reduced representations are considered, it becomes clear why we think that the SVD-reduced space covers the "latent" semantic dimensions of the word-document space.

word	dim_1	dim_2
orange	0.44	−0.29
banana	0.12	−0.33
melon	0.47	−0.51
apple	0.70	0.35
windows	0.26	0.64

Table 3.2: U

2.16	0
0	1.59

Table 3.3: S

doc.	dim_1	dim_2
d_1	0.74	−0.28
d_2	0.27	−0.52
d_3	0.20	−0.18
d_4	0.44	0.62
d_5	0.32	0.21
d_6	0.12	0.40

Table 3.4: V

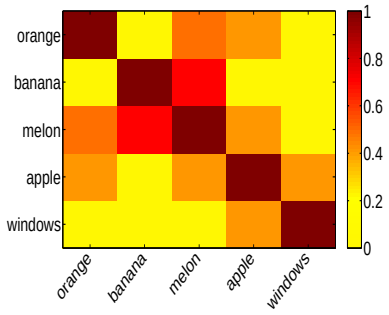


Figure 3.1: *Original word similarities.*

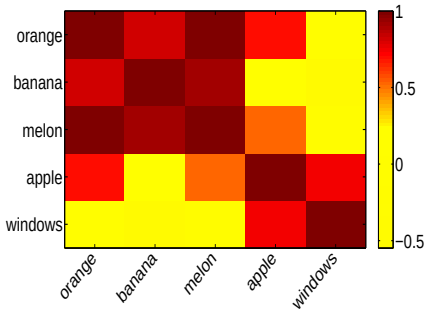


Figure 3.2: *"Latent" word similarities.*

Figure 3.1 shows the word-word similarities given by the cosine similarities of word vectors in the original co-occurrence matrix W. Figure 3.2 shows the word-word similarities given by the cosine similarities of the word representations $u_i S$ in the

latent semantic space. It can be seen that the "latent" semantic space consists of two word clusters, one around the meaning "fruit" and one around the meaning "operating systems". It also makes words similar (*banana*/*orange*) that do not appear in the same document. This similarity cannot be derived from the vectors in the original space.

For document-document similarities shown in Figure 3.3 and 3.4 one can see that documents are similar in the "latent" semantic space that do not have words in common. These similarities cannot be derived directly from the original space. Furthermore one can see the clustering of the document space into two clusters d_1-d_3 and d_3-d_6.

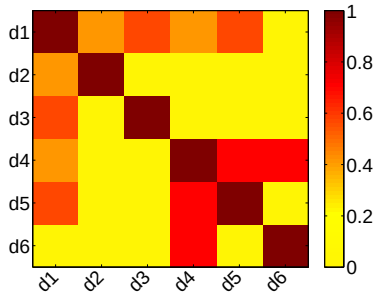


Figure 3.3: *Original document similarities.*

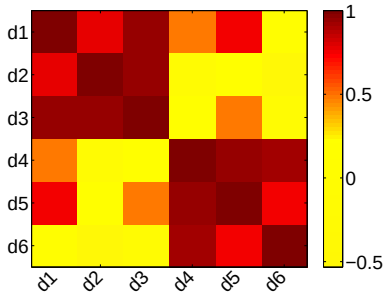


Figure 3.4: *"Latent" document similarities.*

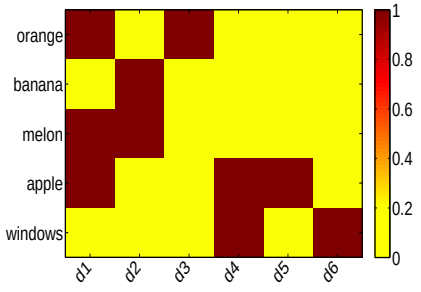


Figure 3.5: *Original word-document similarities.*

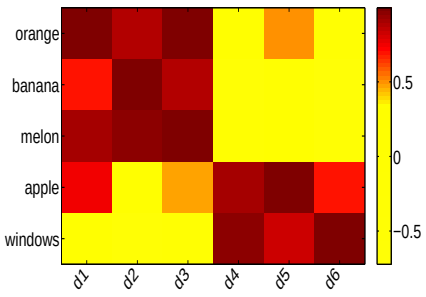


Figure 3.6: *"Latent" word-document similarities.*

A final advantage of LSA is that it allows for a comparison of words and documents. This is especially useful for language modeling where one wants to predict a word from a context/document. The cosine similarity is defined between the word

vector $u_i S^{\frac{1}{2}}$ and the document vector $v_j S^{\frac{1}{2}}$ (Definition 4.8). The reason for this slightly different representation of words and documents, compared to word-word and document-document similarities is explained in Section 4.2.3.

The only way to derive word-document similarities from the original co-occurrence matrix W is to take the matrix itself. A word is then similar to a document if it appears in the document. The similarity matrix derived in this way is shown in Figure 3.5. In the LSA space however similarities can be derived between words and documents, although the word never appeared in the document. The word-document similarities in the "latent" semantic space are shown in Figure 3.6.

3.3.2 Probabilistic latent semantic analysis (PLSA)

PLSA models derive the probabilities between a word and a document or history of words directly, without estimating a similarity first. These models were first applied by Gildea and Hofmann (1999) to language modeling. The basic idea underlying these models is to introduce a latent topic variable.

For estimating the joint probability of a word w, document d, and topic t, one can use the multiplication rule to derive the joint document, word and topic probability.

Definition 3.12 *Joint document, word and topic probability*

$$P(d, w, t) = P(d)P(t \mid d)P(w \mid t, d)$$

With the additional assumption that a word is only dependent on the topic this becomes

Definition 3.13 *Joint document, word and topic probability (word-topic dependence)*

$$P(d, w, t) = P(d)P(t \mid d)P(w \mid t) \ .$$

In the same way one can derive

Definition 3.14 *Joint topic, document and word probability*

$$P(t, d, w) = P(t)P(d \mid t)P(w \mid d, t) = P(t)P(d \mid t)P(w \mid t)$$

which is more symmetric because d and w are only dependent on t (Hofmann, 1999). The graphical model (Cowell, 1998) representations of the models taken from Hofmann (1999) are given in Figure 3.7.

When marginalizing over the variable t one gets the final definition of the PLSA model, with t as the latent topic variable. For Definition 3.14 this yields

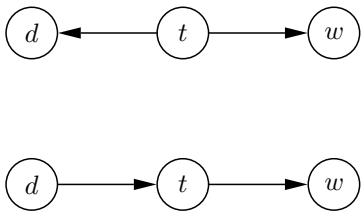


Figure 3.7: *Graphical model representation of PLSA models.*

Equation 3.1 *Joint document and word probability*

$$P(d, w) = \sum_{t \in T} P(t)P(d \mid t)P(w \mid t) \ .$$

Then the model can be estimated with the *Expectation Maximization* (EM) algorithm (Dempster *et al.*, 1977). The expectation step of the EM algorithm, the posterior probabilities for the latent topic variable t are estimated as

Equation 3.2 *Posterior topic probability*

$$P(t \mid d, w) = \frac{P(t, d, w)}{P(d, w)} \ .$$

Using Definition 3.14 and Equation 3.1 this expands to the E-step equation

Equation 3.3 *E-step equation for posterior topic probability*

$$P(t \mid d, w) = \frac{P(t)P(d \mid t)P(w \mid t)}{\sum_{t' \in T} P(t')P(d \mid t')P(w \mid t')} \ .$$

The maximization of the parameters $P(w \mid t)$, $P(d \mid t)$, and $P(t)$ in the maximization step (M-step) is given by the following and similar formulae for the other parameters (Hofmann, 1999).

Equation 3.4 *M-step equation for* $P(w \mid t)$

$$P(w \mid t) \propto \sum_{d \in D} n(d, w)P(t \mid d, w)$$

where $\propto$ signifies that the formula has to be normalized over all words and $n(d, w)$ are the counts for word w in document d.

To be applicable to language modeling one has to introduce a similar concept as the pseudo-document for LSA models, to be able to predict a word given the history of previous words. If the document variable d is replaced by the history variable h in Definition 3.13 this yields

Equation 3.5 *Joint history, word and topic probability*

$$P(h, w, t) = P(h)P(t \mid h)P(w \mid t).$$

With marginalization over t and the multiplication rule one gets

Equation 3.6 *Joint history and word probability*

$$\begin{aligned} P(h)P(w \mid h) &= P(h, w) \\ &= \sum_{t \in T} P(h)P(t \mid h)P(w \mid t) \\ &= P(h) \sum_{t \in T} P(t \mid h)P(w \mid t) \end{aligned}$$

The conditional probability of a word given its history is thereby given by (Gildea and Hofmann, 1999)

Definition 3.15 *PLSA for language modeling*

$$P(w \mid h) = \sum_{t \in T} P(t \mid h)P(w \mid t)$$

such that $P(h)$ does not have to be estimated. The parameters $P(t \mid h)$ have to be computed online as the length of the history increases. Therefore an online EM algorithm as discussed in Neal and Hinton (1998) can be used.

3.4 WordNet-based LM

Several WordNet-based LMs are discussed in Chapter 5. These models are based on measures of lexical semantic relatedness. 'Semantic relatedness' is here used as a general term, which can be divided into 'semantic similarity' and 'semantic distance'. If a word w_1 is semantically closer to w_2 than to w_3 the similarity to w_2 will be higher in a similarity measure and the distance to w_2 will be smaller in a distance measure. Pedersen *et al.* (2004) uses the terms 'semantic similarity' and 'semantic relatedness' for measures within and across POS boundaries, respectively. A wide variety of applications like word sense disambiguation, information extraction, and

word prediction as well as many relatedness measures are described in Budanitsky and Hirst (2001).

Kozima and Ito (1995) define the distance between two words in a given context. For the definition of the basic semantic distance a semantic network is constructed from the *Longman Dictionary of Contemporary English* (LDOCE). The nodes of the network are the words in the dictionary. A link between two nodes w_1 and w_2 is inserted if w_1 appears in the definition of w_2.

As Budanitsky and Hirst (2001) showed this definition can be easily extended such that it can be applied to a word and a set of words in a given context.

Definition 3.16 *Word-sentence distance*

$$\mathrm{dist}_{\mathrm{word}}(w, S, C) = \frac{1}{\mid S \mid} \sum_{w' \in S} \mathrm{dist}(w, w', C)$$

Such a definition is useful in cases where the sentence or utterance context S contains additional information to the context C. Suppose that the semantic similarity between 'fish' and 'school' and other words in C shall be computed. This similarity will be higher if the sentence contains 'coral reef', because in this case it can be assumed that 'school' is used in the seldom used sense of "large group of fish". This means that the similarity between 'fish' and 'school' will be lower than the similarity between 'fish' and 'school' in the context of 'coral reef'

Table 3.5 shows example word-context distances $\mathrm{dist}_{\mathrm{ctxt}}(w, C)$ given by Kozima and Ito (1995) which are defined as the sum of distances $\mathrm{dist}(w, w', C)$ of all words w' in the context. There are two senses of the word 'tour' in this example ("tour_1" and "tour_2").

w	$\mathrm{dist}_{\mathrm{ctxt}}(w, C)$
"bus_1"	0.100833
"$\mathrm{scenery}_1$"	0.112169
"tour_2"	0.122133
"tour_1"	0.128796
"abroad_1"	0.155860
"$\mathrm{tourist}_1$"	0.159336

Table 3.5: *6 highest word-context distances for* $C = \{\mathrm{bus}, \mathrm{scenery}, \mathrm{tour}\}$.

Since the lexical semantic relatedness measures used in this work give a definition of the semantic relatedness between two words ($\mathrm{rel}(w, w')$) the relatedness of a word and a context can be defined directly as

Definition 3.17 *Word-context relatedness*

$$\mathrm{rel}_{\mathrm{word}}(w, C) = \frac{1}{\mid C \mid} \sum_{w_i \in C} \mathrm{rel}(w, w_i) \ .$$

Starting from this basic definition of semantic relatedness between two concepts, more complex relatedness measures can be defined. WordNet-based relatedness measures are either graph-based, or text-based, or both. Examples of all three types of measures are given in Section 5.2.

3.4.1 Graph-based WordNet relatedness measures

With graph-based relatedness measures the relatedness between two words or senses is computed by their position in the WordNet graph. The shortest path between two senses is one possible distance measure. The longer the path the longer the distance between the senses.

Another type of measure computes information content from a corpus, which assigns nodes that are lower in the hierarchy, a higher information content. Then the information content of the *Least Common Subsumer* (LCS) of two senses A and B, which is defined as *the most specific concept that is an ancestor of both A and B* (Pedersen *et al.*, 2004) is taken as the similarity between two senses (Resnik, 1995).

The main disadvantage of these models is that they cannot be applied across POS boundaries in an intuitive way. Of course there is the possibility to simply connect the top nodes of say the verb and noun hierarchy. Chapter 5 shows that in language modeling it is best to use only these measures for nouns.

3.4.2 Text-based WordNet relatedness measures

Text-based relatedness measures are defined on the glosses/definitions of word senses. The WordNet (Fellbaum, 1998, p. 8) graph is organized into synonym sets that are linked to another as shown in Figure 5.1. Additionally WordNet contains a definition for each sense of a word. For the senses of the noun 'gold' these are

(3.1)
1. gold – coins made of gold
2. amber, gold – a deep yellow color; *an amber light illuminated the room*; *he admired the gold of her hair*
3. gold, Au, atomic number 79 – a soft yellow malleable ductile (trivalent and univalent) metallic element; occurs mainly as nuggets in rocks and alluvial deposits; does not react with most chemicals but is attacked by chlorine and aqua regia
4. gold – great wealth; *Whilst that for which all virtue now is sold, and almost every vice–almighty gold*–Ben Jonson

5. gold – something likened to the metal in brightness or preciousness or superiority etc.; *the child was as good as gold*; *she has a heart of gold*

Let X_i and X_j be the set of content words that are contained in the definition of a sense of w_i and w_j, respectively. Then the simplest similarity metric is the so called 'matching coefficient' given by $X_i \cap X_j$. By normalizing over the number of words in the glosses and applying certain smoothing functions one can define more similarity metrics based on the common words of the glosses.

Intuitively it is clear that an n-phrase-overlap is more significant than a $n-1$-phrase-overlap. X_i and X_j have an n-phrase-overlap if they have a common phrase of length n. Higher phrase-overlaps should produce higher similarities. This is one idea that can be taken into account in an adapted text-based measure (Banerjee and Pedersen, 2003).

3.4.3 Hybrid WordNet relatedness measures

Hybrid semantic relatedness measures like the one proposed in Banerjee and Pedersen (2003) use information from the WordNet graph and the glosses/definitions.

If the similarity between two senses has to be estimated, a text-based word or phrase-overlap measure is used that additionally takes neighboring nodes in the graph (graph-based) into account. So not only the phrase-overlap of the two sense definitions is taken into account, but also the phrase-overlaps of the neighboring sense definitions.

Thereby it can be used across POS boundaries and take into account information that is contained in the WordNet graph. It is also possible to train the measures for different types of relations in WordNet and add weights for relation types.

The advantage of text-based and hybrid measures is that they can be applied across POS boundaries. Chapter 5 shows that it is best to use only hybrid measures for verbs and adjectives in language modeling.

3.5 Maximum entropy LM

The maximum-entropy framework was first applied to language modeling by Della Pietra *et al.* (1992) and adopted by Rosenfeld (1994, 1996). Conditional maximum entropy models have the form

Definition 3.18

$$P(w \mid h) = \frac{1}{Z(h)} \exp(\sum_i \lambda_i f_i(w,h))$$

where $f_i(w,h)$ is a discrete or continuous feature defined for a word and its history, $Z(h)$ is a normalization term, and λ_i are the parameters. These models can also be

defined as whole-sentence models where efficient estimation and sampling methods are essential for model estimation (Schofield, 2006).

With these kinds of models it is possible to incorporate multiple types of linguistic features, like n-gram features and LSA-features (Deng and Khudanpur, 2003) into a language model. The advantage of the maximum entropy language modeling framework is that any type of feature can be incorporated. Another viewpoint on the maximum entropy framework is to see it as a smoothing framework that models unseen data using a minimal number of constraints given by the data seen so far.

The main challenge for maximum entropy language modeling in our view is to find good sets of features and good feature combinations such that the combination of features increases the model performance. Deng and Khudanpur (2003) showed how n-gram features and LSA features can be combined and that this combination can improve over an n-gram language model on conversational speech data in terms of perplexity and WER.

Features can either have discrete or continuous values. As an LSA feature one could take the cosine similarity between words and pseudo-document histories in a training corpus defined as

Definition 3.19

$$f_{\text{LSA}}(w,h) = K_{\text{sim}}(w,h) = \frac{uSv^T}{||uS^{\frac{1}{2}}|| \cdot ||vS^{\frac{1}{2}}||}$$

as suggested by Deng and Khudanpur (2003). $uS^{\frac{1}{2}}$ represents w and $vS^{\frac{1}{2}}$ represents h. Of course the pseudo-document history h is different for each word in the training corpus such that this method – taken literally – yields as many parameters as words in the training corpus. While it is possible to have a separate parameter for each n-gram feature for example, it is necessary to perform some parameter tying in the case of LSA features. Otherwise a separate parameter would be needed for each word-history combination. Parameter tying means here that word-history combinations are grouped together, such that only one parameter per group is needed.

One possibility for parameter tying suggested by Deng and Khudanpur (2003) is to use a partitioning of the document space achieved through clustering. If the document space has dimension r and we have some clustering of the $\mathbb{R}^r$ space such that h' represents one cluster in this clustering, a binary LSA feature can be defined as

Definition 3.20

$$\hat{f}_{\text{LSA}}(w,h) = \begin{cases} 1 & \text{if } K_{\text{sim}}(w,h') > \eta, \\ 0 & \text{otherwise.} \end{cases}$$

where h' is the cluster center that is closest to the pseudo-document history h.

The usefulness of this method lies in the possibility to perform the clustering on the LSA space using the cosine document-document similarity (Definition 4.7) and standard clustering algorithms like k-means clustering (MacQueen, 1967). To also cover short-term syntactic information, one has to combine the long-term LSA features with n-gram or similar features.

3.6 Discussion

The main reason to choose LSA and WordNet-based models for this thesis is their ability to include a longer history than n-gram models and to go beyond the sentence context. All other models except PLSA models can either not include a longer history, or they do not go beyond the sentence context.

n-gram models (word-based, class-based, or POS-based) cannot include a long history and cannot go beyond the sentence context. Syntactically or semantically structured LMs can include a longer history but cannot go beyond the sentence context.

LSA, PLSA, and WordNet-based model can include a long history and can go beyond the sentence context. We think that these two properties are important in language modeling for meetings. PLSA models are not considered since they are similar to LSA models. In Chapter 4 and 5 we will investigate if the models can improve ASR for meetings by using a longer history than n-gram models and reaching beyond the sentence context.

4 Latent semantic analysis (LSA) based language models in automatic speech recognition (ASR) for multi-party meetings

> 43. Man kann für eine große Klasse von Fällen der Benützung des Wortes 'Bedeutung' - wenn auch nicht für alle Fälle seiner Benützung - dieses Wort so erklären: Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache (Wittgenstein, 1984, p. 262).[1]

4.1 Introduction

This chapter[2] contains a description of work on *Latent Semantic Analysis* (LSA) based language modeling in *Automatic Speech Recognition* (ASR) for multi-party meetings. Multi-party meetings involve two or more speakers that are engaged in a more or less interactive conversation. Examples of multi-party meetings are regular meetings, talks, and discussions. In this work regular meetings are used as experimental data.

Word-based n-gram models are a popular and fairly successful paradigm in language modeling. With these models it is however difficult to model long distance dependencies which are present in natural language (Chelba and Jelinek, 1998).

LSA defines a semantic similarity space using a training corpus. This semantic similarity can be used for dealing with long distance dependencies, which are an inherent problem for traditional word-based n-gram models. Since LSA models adapt dynamically to topics, and meetings have clear topics, this thesis conjectures that these models can improve speech recognition accuracy on meetings.

LSA maps a corpus of documents onto a semantic vector space. Long distance dependencies are modeled by representing the context or history of a word and the word itself as vectors in this space. The similarity between these two vectors is used to predict the word given the context. Since LSA models the context as a bag-of-words it has to be combined with n-gram models to include word-sequence statistics of the short-span history. Language models that combine word-based n-gram models with LSA models have been successfully applied to conversational speech recognition (Deng and Khudanpur, 2003) and to the Wall Street Journal recognition task (Bellegarda, 2000a,b). In this chapter the LSA approach is applied to ASR for multi-party meetings.

[1] 43. For a large class of cases – though not for all – in which we employ the word 'meaning' it can be defined thus: the meaning of a word is its use in the language (Wittgenstein, 2001, p. 18).

[2] Parts of the content of this chapter were first published in Pucher and Huang (2005); Pucher *et al.* (2006a,b).

Furthermore a method is provided to combine multiple LSA models, such that the same training data that is used for the n-gram models can be used.

Due to the sparseness of available data for meeting language modeling it is important to combine meeting LSA models that are trained on relatively small corpora with background LSA models that are trained on larger corpora. In this case the meeting domain is the adaptation domain (Bellegarda, 2004) and there are multiple background domains from broadcast news to web data (Table 4.7).

This chapter presents perplexity and *Word-Error-Rate* (WER) results for LSA models for meetings. Results for models trained on a variety of corpora including meeting data and background domain data are presented. Furthermore combinations of multiple LSA models and word-based n-gram models are investigated.

It is shown that meeting and background LSA models can improve over the baseline n-gram models in terms of perplexity and that some background LSA models can significantly improve over the n-gram models in terms of WER. For the combination of multiple LSA models no improvement can be reported.

Additionally to perplexity and WER results an extensive analysis of LSA models is conducted. This analysis covers the optimization of LSA model parameters necessary for the interpolation of multiple LSA models as well as a comparison of LSA and cache-based models. This comparison shows that the former contain more semantic information than is contained in the repetition of word forms alone.

4.2 LSA-based language models

4.2.1 Constructing the semantic space

In LSA first a training corpus is encoded as a word-document co-occurrence matrix W (using weighted term frequency). This matrix has high dimension and is highly sparse. Let $\mathcal{V}$ be the vocabulary with $|\mathcal{V}| = M$ and $\mathcal{T}$ be a text corpus containing N documents.

Let c_{ij} be the number of occurrences of word i in document j, c_i the number of occurrences of word i in the whole corpus, that is $c_i = \sum_{j=1}^{N} c_{ij}$, and c_j the number of words in document j, that is $c_j = \sum_{i=1}^{M} c_{ij}$. The elements of W are given by

Definition 4.1 *Encoding of word-document co-occurrence matrix*

$$[W]_{ij} = (1 - \epsilon_{w_i}) \frac{c_{ij}}{c_j} \ .$$

$\frac{c_{ij}}{c_j}$ is the ratio of the count of a word in a document c_{ij} and the total number of words in that document c_j. In this way a given count for a word gives a higher weight in a smaller document. So the length of a document is taken into account.

The informativeness of a word shall also be taken into account. Content words are more informative than function words. From a content word it should be possible to make a guess on the topic of a document. Since only counts derived from a corpus and no prior linguistic knowledge is used, the distinction between content and function words depends on the corpus. A content word is a word that only appears in some documents, while a function word appears in many or most documents. The un-informativeness ϵ_{w_i} of a word w_i is defined as

Definition 4.2 *Word un-informativeness (entropy)*

$$\epsilon_{w_i} = -\frac{1}{\log_2 N} \sum_{j=1}^{N} \frac{c_{ij}}{c_i} \log_2 \frac{c_{ij}}{c_i} \ .$$

ϵ_w is used as a short-hand for ϵ_{w_i}. Informative words have a low value of ϵ_w. For words appearing everywhere in the corpus $\frac{c_{ij}}{c_i}$ is almost uniform and therefore ϵ_w is high and close to 1. For words that appear in a more content-word like fashion the value is lower.

Then a semantic space with much lower dimension is constructed using Singular Value Decomposition (SVD) (Deerwester *et al.*, 1990; Berry, 1993).

Definition 4.3 *Singular value decomposition*

$$W \approx \hat{W} = USV^T$$

For some order $r \ll \min(m, n)$, m is the number of words, n is the number of documents, U is a $m \times r$ left singular matrix, S is a $r \times r$ diagonal matrix that contains r singular values, and V is a $n \times r$ right singular matrix. The vector $u_i S$ represents word w_i, and $v_j S$ represents document d_j.

4.2.2 Pseudo-document representation

The concept of a pseudo-document $\tilde{\mathrm{d}}_{t-1}$ using the word vectors of all words preceding w_t $(w_1, \ldots, w_{t-1})$ is needed because the model is used to compare words with word histories. It is very likely that these histories are not present in the training corpus. Therefore the histories are encoded as pseudo-documents. In the construction of the pseudo-document a decay parameter $\delta < 1$ is included that renders words closer in the history more significant.

As a matrix column that can be added to the matrix $\hat{W}$ the pseudo-document is defined as

Definition 4.4 *Inductive definition of pseudo-document*

$$\tilde{d}_t = \frac{t-1}{t} \tilde{d}_{t-1} + \frac{1 - \epsilon_{w_t}}{t} e_{w_t}$$

where ˜ denotes that the document is not in the original matrix and e_{w_t} is a word selection vector with 0 at all places except for the position w_t where it is 1. Here it is desired to find an expression of the pseudo-document that is equal to the representation of the other documents in the semantic space.

These documents are represented as $v_t S$, so $\tilde{\mathrm{v}}_t S$ is the desired vector. Since $\tilde{\mathrm{v}}_t S$ is not in the original matrix[3] it has to be constructed from the word vectors u_i of the matrix U. Since it is known that $\tilde{\mathrm{v}}_t S = \tilde{\mathrm{d}}_t^T U$, this fact can be used in conjunction with Definition 4.4 to find the desired pseudo-document representation.

Lemma 4.1 *Representation of pseudo-document in semantic space*

$$\tilde{v}_t S = \frac{t-1}{t}\tilde{v}_{t-1}S + \frac{1-\epsilon_{w_t}}{t}u_{w_t}$$

Proof 4.1

$$\begin{aligned}
\tilde{v}_t S &= \tilde{d}_t^T U \\
&= (\frac{t-1}{t}\tilde{d}_{t-1} + \frac{1-\epsilon_{w_t}}{t}e_{w_t})^T U \qquad \text{(Definition 4.4)} \\
&= \frac{t-1}{t}\tilde{d}_{t-1}^T U + \frac{1-\epsilon_{w_t}}{t}e_{w_t}^T U \\
&= \frac{t-1}{t}\tilde{v}_{t-1}S + \frac{1-\epsilon_{w_t}}{t}u_{w_t}
\end{aligned}$$

Now the already mentioned decay $\delta < 1$ has to be added that is applied to all words except the current word. So the final definition of the pseudo-document is

Definition 4.5 *Decaying representation of pseudo-document*

$$\tilde{v}_t S = \delta\frac{t-1}{t}\tilde{v}_{t-1}S + \frac{1-\epsilon_{w_t}}{t}u_{w_t}\ .$$

The optimization of the decay parameter δ is described in Section 4.4.5. This decay parameter can be set according to the task while $\frac{t-1}{t}$, which also has a decaying effect defines a fixed sequence of values.

4.2.3 LSA probability

In this semantic space one can define word-word, document-document, and word-document similarities. Word-word similarities can be used for the clustering of words, document-document similarities can be used for document clustering, and similarities between words and documents can be used in language modeling. The cosine similarity between words is defined as

[3] To denote this fact the ˜ notation is used.

Definition 4.6 *Cosine similarity between words*

$$K_{\text{simword}}(w_i, w_j) = \frac{u_i S^2 u_j^T}{||u_i S|| \cdot ||u_j S||}$$

which is simply the cosine of the angle between the vectors $u_i S$ (representing w_i) and $u_j S$ (representing w_j). $(u_i S)(u_j S) = u_i S^2 u_j^T$ since $vS = Sv^T$ for a diagonal matrix S and a vector v. The cosine similarity measures how close the two words are in this space.

The matrix $\hat{W}\hat{W^T}$ characterizes all co-occurrences between words. Therefore the association between two words can be derived from the (i, j) cell of this matrix. This cell is equal to the dot product $(u_i S)(u_j S)$.

Theorem 4.1 *Word association space*

$$\hat{W}\hat{W^T} = USUS$$

Proof 4.2

$$\begin{aligned}
\hat{W}\hat{W^T} &= USV^T(USV^T)^T && \text{(Definition 4.3)} \\
&= USV^T(US(V^T))^T && \text{(matrix associativity)} \\
&= USV^T(V^T)^T S^T U^T && ((US)^T = S^T U^T) \\
&= USV^T V S^T U^T && ((V^T)^T = V) \\
&= USV^T V S U^T && (S^T = S \text{ for diagonal matrices}) \\
&= USSU^T && (V^T V = I \text{ for orthogonal matrices}) \\
&= USUS && (SU^T = US \text{ for diagonal matrices})
\end{aligned}$$

In the same way one can show that $\hat{W^T}\hat{W} = VSVS$. Between documents the cosine similarity is therefore defined in the same way:

Definition 4.7 *Cosine similarity between documents*

$$K_{\text{simdoc}}(d_i, d_j) = \frac{v_i S^2 v_j^T}{||v_i S|| \cdot ||v_j S||}$$

The cosine similarity between words and documents, which is used for predicting a word given a document context, is defined slightly different as

Definition 4.8 *Cosine similarity between words and documents*

$$K_{\text{sim}}(w_i, d_j) = \frac{u_i S v_j^T}{||u_i S^{\frac{1}{2}}|| \cdot ||v_j S^{\frac{1}{2}}||} \ .$$

The idea behind this definition is that the relation between a word w_i and a document d_j is given by the (i, j) cell of the matrix $\hat{W}$. Since $\hat{W} = USV^T$ this cell value is equal to the dot product of $u_i S^{\frac{1}{2}}$ and $S^{\frac{1}{2}} v_j^T$, which is equal to the dot product of $u_i S^{\frac{1}{2}}$ and $v_j S^{\frac{1}{2}}$. Therefore the similarity of a word and a document is defined as the cosine similarity between these two vectors.

Since a probability is needed for the integration with conditional n-gram models, this similarity is converted into a probability by normalizing it. Because this conversion is seen as a weakness of LSA models, other methods have been proposed that derive the probabilities directly (Gildea and Hofmann, 1999). According to Coccaro and Jurafsky (1998), the small dynamic range of the similarity function is extended by introducing a similarity exponent parameter γ. Our experiments also show that the optimization of γ can lower the perplexity. In Section 4.4.4 the effect of this parameter is analyzed.

The conditional probability of a word w_t given a pseudo-document $\tilde{\mathrm{d}}_{t-1}$ is defined as

Definition 4.9 *LSA probability*

$$P_{\mathrm{LSA}}(w_t|\tilde{d}_{t-1}) = \frac{[K_{\mathrm{sim}}(w_t, \tilde{d}_{t-1}) - K_{\mathrm{min}}(\tilde{d}_{t-1})]^\gamma}{\sum_w [K_{\mathrm{sim}}(w, \tilde{d}_{t-1}) - K_{\mathrm{min}}(\tilde{d}_{t-1})]^\gamma}$$

where $K_{\mathrm{min}}(\tilde{\mathrm{d}}_{t-1}) = \min_w K(w, \tilde{\mathrm{d}}_{t-1})$ to make the resulting similarities non-negative. A small offset is added to $K_{\mathrm{sim}}(w_t, \tilde{\mathrm{d}}_{t-1})$ such that no zero probabilities are computed in the case that $K_{\mathrm{sim}}(w_t, \tilde{\mathrm{d}}_{t-1}) = K_{\mathrm{min}}(\tilde{\mathrm{d}}_{t-1})$

4.2.4 Combining LSA and n-gram models

In the adaptation of statistical language models one tries to find a robust estimate of the language model probability for sequences of words given a history of words (Bellegarda, 2004). For LSA-based models this estimate is a combination of LSA and n-gram models. The long-term history is covered by the LSA model, the short-term word sequence is covered by the n-gram model.

For the interpolation of the word-based n-gram models and the LSA models the methods defined in Table 4.1 are used. λ is a fixed constant interpolation weight, and $\propto$ denotes that the result is normalized by the sum over the whole vocabulary. λ_w is a word-dependent parameter defined as (Deng and Khudanpur, 2003)

Definition 4.10 *Word-dependent* λ

$$\lambda_w = \frac{1 - \epsilon_w}{2} \ .$$

λ_w is a short-hand for λ_{w_i}. This definition ensures that the n-gram model gets at least half of the weight. λ_w is higher for more informative words.

Model	Definition
n-gram (baseline)	$P_{n\text{-gram}}$
Linear interpolation (LIN)	$\eta P_{\text{LSA}} + (1-\eta) P_{n\text{-gram}}$
Similarity modulated n-gram interpolation (SIMMOD)	$\propto (K_{\text{sim}} - K_{\text{min}}) P_{n\text{-gram}}$
Information weighted geometric mean interpolation (INFG)	$\propto P_{\text{LSA}}^{\lambda_w} P_{n\text{-gram}}^{1-\lambda_w}$

Table 4.1: *Interpolation methods.*

Three different methods (Coccaro and Jurafsky, 1998; Deng and Khudanpur, 2003) for the interpolation of n-gram models and LSA models are applied. The "information weighted geometric mean", the "similarity modulated n-gram" and simple "linear interpolation". The "information weighted geometric mean" interpolation represents a log-linear interpolation (Klakow, 1998) of normalized LSA probabilities and the standard n-gram, weighted by λ_w. The "similarity modulated n-gram" interpolation uses K_{sim} and K_{min} directly, without normalizing first.

4.2.5 Combining LSA models

In a further adaptation multiple LSA models are interpolated. The models are thereby divided into application domain models, which is in our case the meeting model and the background models.

For the combination of LSA models two different approaches are defined here. The first approach is a straightforward generalization of the linear interpolation for multiple LSA models with optimized η_i where $\eta_{n+1} = 1 - (\eta_1 + \ldots + \eta_n)$:

Definition 4.11 *Linear interpolation*

$$P_{\text{lin}} = \eta_1 P_{\text{LSA}_1} + \ldots + \eta_n P_{\text{LSA}_\text{n}} + \eta_{n+1} P_{n\text{-}gram}$$

The second approach is a generalization of the INFG Interpolation with optimized θ_i where $\lambda_w^{(n+1)} = 1 - (\lambda_w^{(1)} + \ldots + \lambda_w^{(n)})$. An INFG type of interpolation was already used by Coccaro and Jurafsky (1998); Deng and Khudanpur (2003) for interpolation of n-gram and LSA models. Here we generalize this method for multiple LSA models

which also results in a generalization of the word-dependent parameters. Thereby global model parameters can be trained for LSA models and the n-gram model.

Definition 4.12 *INFG interpolation*

$$P_{\text{infg}} \propto P_{\text{LSA}_1}^{\lambda_w^{(1)}\theta_1} \dots P_{\text{LSA}_\text{n}}^{\lambda_w^{(n)}\theta_n} P_{n\text{-}gram}^{(1-(\lambda_w^{(1)}+\dots+\lambda_w^{(n)})\theta_{n+1})}$$

The parameters θ_i have to be optimized since the $\lambda_w^{(k)}$ depend on the corpus, such that a certain corpus can get a higher weight because of a content-word-like distribution of w, although the whole data does not well fit the meeting domain. One such optimization algorithm will be discussed later in Section 4.2.7. In general the λ_w values are higher for the background domain models than for the meeting models. But taking the n-gram mixtures as an example the meeting models should get a higher weight than the background models. For this reason the λ_w of the background models have to be lowered using θ.

To ensure that the n-gram model gets a certain part κ of the distribution, $\lambda_w^{(k)}$ is defined for word w and LSA model LSA_k as

Definition 4.13

$$\lambda_w^{(k)} = \frac{1 - \epsilon_w^{(k)}}{\frac{n}{1-\kappa}}$$

where $1 - \epsilon_w^{(k)}$ is the informativeness of word w in LSA model LSA_k as defined in (4.2) and n is the number of LSA models. This is a generalization of definition (4.10). Through the generalization it is also possible to train κ, the minimum weight of the n-gram model.

For the INFG interpolation the model parameters θ_i, the weight of the n-gram model κ, and the γ exponent for each LSA model have to be optimized. All parameters are optimized using the gradient descent algorithm (Section 4.2.7).

4.2.6 Perplexity

The performance of a statistical language model is determined by the perplexity of the model given some test data. The perplexity is defined as $2^{H(p,m)}$ using the cross-entropy $H(p, m)$ of a language model m and the true probability distribution p of a language L, according to which sequences of words are drawn. The cross-entropy is defined as

Definition 4.14 *Cross entropy 1*

$$H(p, m) = -\lim_{n\to\infty} \frac{1}{n} \sum_{W \in L} p(w_1, \dots, w_n) \log_2 m(w_1, \dots, w_n)$$

where W are the sequences of length n in the language L (Jurafsky and Martin, 2000, p. 227). To compute this, one would have to know the true probability distribution p of the language L. According to the Shannon-McMillan-Breiman theorem (Cover and Thomas, 1991, p. 475) for stationary and ergodic processes this formula is equivalent to

Definition 4.15 *Cross entropy 2*

$$H(p, m) = \lim_{n \to \infty} -\frac{1}{n} \log_2 m(w_1, \ldots, w_n) \ .$$

Since $H(p)$, the cross-entropy of the true distribution is always smaller than $H(p, m)$ the performance of a language model can be measured by its closeness to the true cross-entropy. In practice, sequences of infinite length are not encountered. So the cross-entropy and perplexity of a model is always approximated and restricted to the "perplexity of a test set".

4.2.7 Perplexity optimization with gradient descent

"Gradient descent" is an algorithm for approximating the local minimum of a function. For this work the perplexity of combined models shall be optimized by optimizing the η_i's in Definition 4.11 and the θ_i's in Definition 4.12. The exponent γ in Definition 4.9 and the decay parameter δ in Definition 4.5 are optimized using the same method.

The functions to optimize are

$$f_{\text{lin}}(\text{LSA}_1, \ldots, \text{LSA}_\text{n}, n\text{-gram}, \eta_1, \ldots, \eta_n)$$

$$f_{\text{infg}}(\text{LSA}_1, \ldots, \text{LSA}_\text{n}, n\text{-gram}, \theta_1, \ldots, \theta_{n+1})$$

where the η value for the n-gram is given by $1 - (\eta_1 + \ldots + \eta_n)$. f_{lin} is the perplexity of the linearly interpolated models $\text{LSA}_1, \ldots, \text{LSA}_\text{n}, n$-gram using the interpolation parameters $\eta_1, \ldots, \eta_n, 1-(\eta_1+\ldots+\eta_n)$. f_{infg} is the perplexity of the INFG interpolated models $\text{LSA}_1, \ldots, \text{LSA}_\text{n}, n$-gram using the interpolation parameters $\theta_1, \ldots, \theta_{n+1}$. The models are not changed during optimization, so the functions can be written as

$$f_{\text{lin}}(\eta_1, \ldots, \eta_n)$$

$$f_{\text{infg}}(\theta_1, \ldots, \theta_{n+1})$$

Since the gradient is needed for this method, which involves the derivative of the functions that cannot be determined analytically, it is necessary to approximate the derivative of the functions. The iteration of the θ_i is defined as follows:[4]

[4]The optimization of all other parameters is done in the same way.

Definition 4.16 *Gradient descent*

$$\theta_{i_{j+1}} = \theta_{i_j} - \alpha f'(\theta_{i_j})$$

α is the step size. θ_{i_j} is the value of θ_i at step j. If f is ascending in the point θ_{i_j}, then the derivative $f'(\theta_{i_j})$ is positive and $\theta_{i_{j+1}}$ decreases, otherwise it increases until the function converges, provided that α is chosen small enough. To estimate the function f_{infg} (or the function f_{lin}) one has to introduce another parameter β, the step width that is used to compute the steepness of f_{infg} for some θ_i where the other θ_i's are fixed.

Definition 4.17 *Approximation of perplexity function gradient*

$$f'_{\text{infg}}(\theta_i, \beta) = \frac{f_{\text{infg}}(\theta_i) - f_{\text{infg}}(\theta_i + \beta)}{\beta}$$

Algorithm 4.1 *Perplexity optimization*
OPTIMIZE-PERPLEXITY($f_{\text{infg}}(\theta), \alpha, \beta$)
1 $\Theta \leftarrow \theta$ $p_{old} \leftarrow 0$ $p_{new} \leftarrow 1$
2 **while** $\mid p_{old} - p_{new} \mid > 0.1$
3 **do for** $i \leftarrow 1$ **to** $n + 1$
4 **do** $\Theta_i \leftarrow \theta_i - \alpha\ f'_{\text{infg}}(\theta_i, \beta)$
5 $p_{old} \leftarrow f_{\text{infg}}(\theta)$
6 $p_{new} \leftarrow f_{\text{infg}}(\Theta)$
7 $\theta \leftarrow \Theta$
8 **return** θ

Here again β has to be small enough. The experiments in Section 4.4.2 show that it is sufficient to choose a value for the parameter between 0.05 and 0.1. The assumption underlying this approximation is that the function is a straight line between θ_i and $\theta_i + \beta$. If this assumption is false and there is a local minimum between θ_i and $\theta_i + \beta$ this local minimum is missed.

Algorithm 4.1 shows the complete algorithm for the optimization of the θ's, that converges if the perplexity difference is below 0.1. Using this method the model parameters γ and δ, the interpolation parameters η and θ, and the weight of the n-gram model κ are optimized.

4.2.8 Out-of-vocabulary (OOV) words

LSA-based language models are trained on a corpus that is divided into documents. A document is considered as a bag-of-words by the model. Since word order is of

high importance in languages like English, for these languages these models have to be combined with models that also consider word ordering, like word-based n-gram models or another kind of structured language model.

If the models are trained on the same corpus they can use the same vocabulary. When interpolating multiple LSA models with an n-gram model, the n-gram model is trained on all the data and interpolated with different LSA models that are trained on parts of the data. Then it happens that the n-gram model vocabulary contains a word that is missing in some of the LSA model vocabularies.

A special event that has to be included in the n-gram vocabulary to define the probabilities over sentences and not only over sequences of a certain length, is the end-of-sentence symbol <\s> (Chen and Goodman, 1998). It is not clear how to determine the probability of this event using an LSA model. Since this model rests on a bag-of-word assumption it predicts the end of a sentence from the average number of sentences in a corpus, and does not use any word ordering information. In addition it lowers the probability of the end-of-sentence event compared to the n-gram model, a prediction where the n-gram model is especially useful. For these reasons the end-of-sentence event is not included in the LSA vocabulary and the n-gram alone is used for predicting the end of a sentence.

4.3 Semantics of LSA-based language models

The topic of the 'semantics' of LSA models has already been studied by multiple authors, although their results are not always compatible.

Landauer and Dumais (1997) presents an extensive study on the semantics of LSA. The main conclusions of this study are:

1. LSA is capable of capturing higher-order indirect associations - LSA is capable of capturing higher-order semantic similarities according to the transitivity of similarity which is defined as
    - If A and B are similar and B and C are similar, A and C are similar (even if A and C never appear in the same contexts)
2. LSA has inductive power - 75% of LSA knowledge about words is derived from documents that do not contain the words. To measure this synonymity experiments were conducted with LSA models trained on text containing the synonyms and text not containing them. It was shown that the discriminative power of the model for the synonymity task increases with the size of the training corpus, even when the training corpus did not contain the synonyms.

Ad (1): If semantic similarity is not at least partially transitive, it would not be justified to call the relation a 'similarity'. Transitivity is one defining condition of similarity besides reflexivity and symmetry.

Ad (2): This result is needed if the method is to be used to explain human language learning on an empirical basis. The epistemological problem of the empiricist is how to derive a complex picture of the world by just being exposed to sensory data. Since LSA is capable of deriving most of its word knowledge indirectly by just being exposed to a text corpus (empirical basis) it is possible to learn semantic similarity by limited amounts of data. At least with a corpus that is as limited as the number of experiences a human child has during the process of language acquisition.

Wandmacher (2005) gives a slightly different picture of LSA performance concerning semantics. The author compares LSA performance to the performance of a collocation-based method on the identification of semantic relations. Semantic relations are divided into 'semantic relations', which are lexical semantic relations like synonymy ('has the same sense as') and hypernymy ('is a super ordinate of'), 'morphological relations' that hold if a word is a morphological derivation of another word (*student*, *students*), 'associations' that are intuitive relations of semantic association (*airplane / to land*, *cat / milk*) (Wandmacher, 2005) and 'erroneous relations'.

'Intuitive' means here that the labelers that performed the classification decided that an association relation holds, on the basis that they knew some sample prototypical associations. The classification task was done by two German native speakers. It would clearly be interesting how these native speakers determine the presence of an association. One could try to classify each possible predication as an association, but this definition would be too wide. The examples given by Wandmacher (2005) suggest that associations are more like necessary properties/functions of an object (*airplane / to land*)[5] and stereotypes (*cat / milk*).

The class of 'associations' was introduced when the poor performance of LSA for the other relations was realized (Personal communication with the author). When taking the 5 nearest neighbors of a word the number of non-erroneous relations is only around 50%. No threshold for similarity was used, so this could be one reason for the poor performance.

Further analysis in Wandmacher (2005) shows that LSA performance in capturing the multiple meanings of ambiguous words is worse than the performance of the collocation-based method. The collocative significance $\text{sig}(A, B)$ is defined according to Quasthoff and Wolff (2002) as

Definition 4.18 *Collocative significance*

$$\text{sig}(A, B) = \frac{f_A f_B}{n} - f_{AB} \log_2(\frac{f_A f_B}{n}) + \log_2(f_{AB}!)$$

where f_A is the number of contexts where A occurs and f_{AB} is the number of contexts where A and B co-occur in a paragraph among n paragraphs. In the case of ambiguous

[5] An airplane would not be the kind of thing that it actually is, if it would not have the possibility to land.

words LSA always favors the dominant sense. This shortcoming of LSA models was already known. Hofmann (1999) also argued for other latent variable models like *Probabilistic Latent Semantic Analysis* (PLSA), because they are capable of representing semantically ambiguous words.

4.3.1 Lexical meaning and sentence meaning

Following Lyons (1995, p. 33) we distinguish between lexical or word meaning, sentence meaning, and utterance meaning. Since there are too many semantic concepts, that eventually could be covered by LSA models, the discussion is restricted to a certain concept from each type of meaning. In the context of lexical meaning this is 'synonymy', in the context of sentence meaning 'predication' and 'negation' are discussed. Utterance meaning is not covered. Utterance meaning deals with the role of speech acts (questions, promises, etc.) and discourse relations like conversational implicatures (Grice, 1981). Sentence meaning is at the heart of logical semantics and is mostly dealing with the concept of "truth" and related concepts (negation, predication, quantification, etc.). The notation for words, word forms, and meanings follows the conventions given in Chapter 1.

Synonymy

The idea to define identity via a substitution principle goes back to Leibniz. He argued that "those terms of which one can be substituted for the other without affecting truth are identical" (Ishiguro, 1990, p. 17), or in Latin words "Eadem sunt, quorum unum alteri substitui potest salva veritate". Therefore it is called the 'salva veritate' principle.

In WordNet synonymity is defined in a weaker sense restricted to certain contexts.

> The notion of synonymy used in WordNet does not entail interchangeability in all contexts; by that criterion, natural languages have few synonyms. The more modest claim is that WordNet synonyms can be interchanged in some contexts. To be careful, therefore, one should speak of synonymy relative to a context, but in order to facilitate the discussion this qualification will usually be presupposed, not asserted (Miller, 1998).

It is interesting that this definition does not mention a concept that may not change through the interchange. Without such a concept however the whole definition of interchangeability is meaningless. The problem is that many concepts that can be used in this definition make it circular. Like the definition that two words are synonymous if they have the same meaning in all contexts. This means to define sameness-of-meaning with sameness-of-meaning.

Landauer and Dumais (1997) contains an evaluation on the performance of LSA models on a synonymity test compared to human synonymity judgments. The work shows that LSA has the same performance as human non-native speakers in synonymity judgment tests. This result is encouraging, although it is only valid for non-native human speakers of a language. Native speakers would probably outperform the LSA model on that task. It shows how LSA can explain the capacity of humans to make synonymity judgments.

Predication

The basic concept of logical semantics is the concept of logical consequence. To model logical consequence for classical first-order predicate logic (Quine, 1981) one has to be able to model at least negation, disjunction (Lukasiewicz, 1935), existential quantification, predication, and identity. All other logical connectives and quantifiers can be defined with these concepts. It is not argued that the formal languages of logic are the ideal candidates for an analysis of natural language. But the study of these languages makes clear what is necessary to model the concept of logical consequence or inference. And this concept is surely an important concept of sentence meaning.[6]

Kintsch (2001) discusses a method of taking a predicate (*ran*) and modifying the LSA-space using this predicate such that the semantic similarity between the predicate and its reasonable arguments like *horse* is higher than its similarity to unreasonable arguments like *color*. The modification of the LSA-space thereby provides a comprehension model, while the LSA model itself can be seen as a knowledge base.

Negation

Widdows and Stanley (2003) uses a similar methodology to LSA to derive word meanings as vectors (Schütze, 1998), and defined negation and disjunction on these word vectors. As the underlying logic they used quantum logic (Birkhoff and von Neumann, 1936), not classical logic. The purpose of these definitions is to use the semantic space for *Information Retrieval* (IR).

In classical Boolean logic the disjunction of two events A and B is represented by set union $A \cup B$. It follows that if $a \in A \cup B$ then at least one of the statements $a \in A$, $a \in B$ must hold (Widdows and Stanley, 2003). In quantum mechanics it is possible that this principle is violated (Putnam, 1976). The problem can be solved by replacing the sets by vector subspaces and the set union by the vector sum $+$. Then

[6] Bos and Markert (2005) show how deep and shallow methods can be combined to compute textual entailment, which is the relation when a text logically implies another text. The shallow methods used there rely on a concept of text overlap. Carnap and Bar-Hillel (1952) showed how it is possible to go the other way around, from a rich logical structure to a concept of semantic information. Their classic study is a good starting point to see what is lost during this transformation, and why this transformation is not simply invertible.

the above principle does not hold any longer since there are many elements in $A + B$, which are neither in A nor in B.

4.3.2 Synonymity experiments

To test the performance of LSA-based language models on a synonymity task LSA similarities are compared with synonymity classes given in WordNet. Thereby the two modeling approaches are related to one another.

For this comparison an LSA model on the Fisher conversational speech data (Table 4.7) is trained. The similarity between words in the semantic space is defined as the cosine similarity (Definition 4.6). The vocabulary size of the model is $53,753$.

From this model words are deleted, that only appear once in the corpus. For these words the LSA model provides no semantic. This results in a vocabulary containing $33,933$ items.

Then the intersection of words in the LSA model and in WordNet is taken, without considering word forms. So if *student* and *students* is found in the vocabulary only the base form is included in the common vocabulary. Testing the correlation of metrics for word forms would be another task. Other experiments showed that word forms tend to have high LSA similarity, which is intuitively clear since they are very likely to appear in the same documents.

From this vocabulary words are selected that have at least one synonym in WordNet that is also in the vocabulary. Otherwise the words would form a singleton synonym class, and the LSA performance cannot be measured on such classes. Carrying out these pre-processing steps the size of the vocabulary is further reduced from $33,933$ to $11,519$ items.

Then the synonym classes are extracted according to WordNet synonyms for the reduced vocabulary. A synonym class is defined as the words in the reduced vocabulary that are synonymous regarding one sense. This uses a concept of weak synonymy where it is not required that words are synonymous in all contexts. All senses of both words are selected and the intersection between these senses is taken. If this is not the empty set, the synonyms belong to the respective synonym class.

Synonym class	Words
"source$_{\mathrm{N},9}$"	*reservoir, source*
"source$_{\mathrm{N},7}$"	*author, generator, source*
"source$_{\mathrm{N},6}$"	*germ, seed, source*
"source$_{\mathrm{N},3}$"	*reference, source*
"source$_{\mathrm{N},2}$"	*informant, source*
"source$_{\mathrm{N},1}$"	*beginning, origin, root, source*

Table 4.2: *Synonym classes for the word form* source.

Table 4.2 shows the synonym classes of the word form *source*. "source$_{N,9}$" signifies the 9th sense of the noun "source". It can be seen that a certain word form belongs to multiple synonym classes. It can of course also happen that a word form has multiple POS. In this way the space is further reduced to 9413 synonym classes.

To test the correlation between LSA models and WordNet synonyms, the mean similarity of a word to all other words (excluding the word itself) is computed. This is done for each word in the vocabulary. As the second random variable the mean similarity of a word to all other words in the same synonym classes (excluding the word itself) is computed. For Table 4.2 this is the within-class-similarity of *reservoir, source*, the within-class-similarity of *author, generator, source* and so on.

T-tests for paired samples indicate that the LSA similarities within synonym classes are significantly higher than the similarities over the whole vocabulary ($p < 0.005$, $\#\text{words} = 11,519$). This indicates further that the semantics given by the LSA models can be used for approximating synonymity.

Figure 4.1 shows the LSA similarities for 31 words in the vocabulary belonging to 8 randomly selected synonym classes. The LSA similarities are symmetric since the cosine similarity is symmetric. The highest value (= 1.0) is reached with self-similarity of words shown along the diagonal.

One can see that some very high similarities (> 0.6) are within synonym classes shown in black squares. These are between *option* and *choice* in the synonym class (*choice*, *option*, *pick*, *selection*), between *prohibited* and *forbidden* within (*forbidden*, *out*, *prohibited*, *taboo*) and between *taught* and *schooled* in the synonym class (*instructed*, *schooled*, *taught*). The inclusion of *out* in the (*forbidden*, *out*, *prohibited*, *taboo*) class may seem surprising, but is due to the representation of rare senses in WordNet. The example given for this sense of the adjective *out* is *In our house dancing and playing cards were out.*

But there are also other high similarities namely between *lucky/growing*, *out/wrapped*, *out/choice*, *out/growing*, and *out/lucky*. The high similarities to *out* can be lowered by applying POS tagging. Thereby counts for the adverb *out* (*Get out of here!*) and the preposition *out* (*looking out of the window*) are not added to counts of the adjective *out*. Overall however the similarities within synonym classes are higher than the similarities within the whole vocabulary. The self-similarities are excluded in the computation of the mean similarities.

This result is promising for the application of the models to the task of language modeling. It remains to be seen how strong the impact of these semantic relations is for predicting word sequences.

4.4 Analysis of the models

To gain a deeper understanding of the models, the effects of model parameters are analyzed and LSA models are compared to other similar models. For this analysis

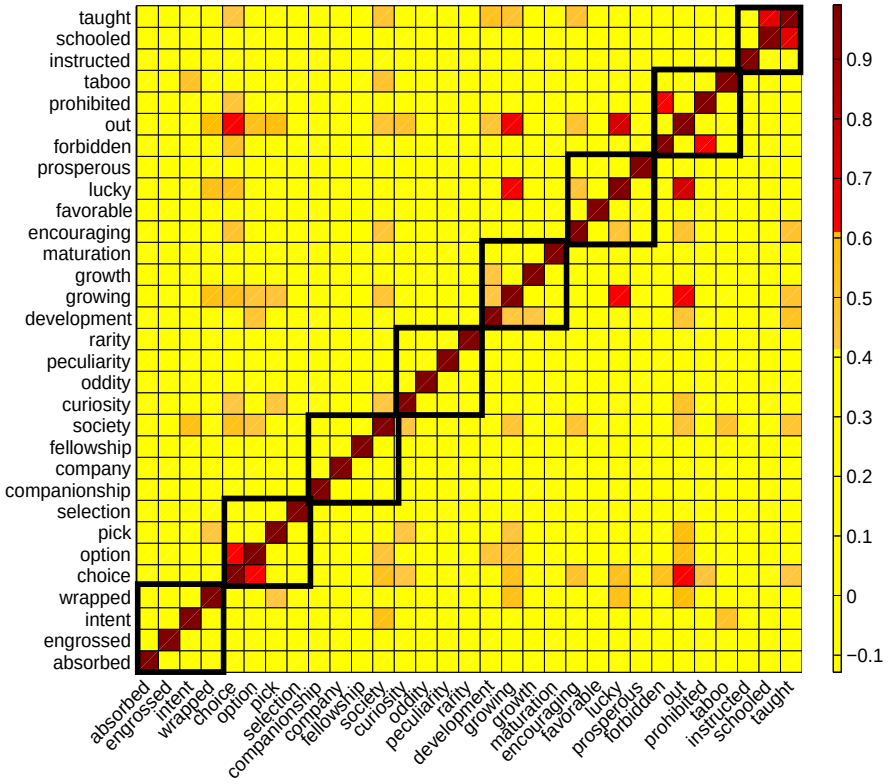


Figure 4.1: *Sample LSA similarities for WordNet synonym classes.*

meeting heldout data containing four ICSI, four *Carnegie Mellon University* (CMU) and four *National Institute of Standards and Technology* (NIST) meetings, is used (Table 4.8). The perplexities and similarities are estimated using LSA and 4-gram models trained on the Fisher conversational speech data and the meeting data (Table 4.7). The models are interpolated using the INFG interpolation method (Table 4.1).

4.4.1 Visualizing the semantic space

Table 4.3 shows sample un-informativeness values ϵ_{w_i} (Definition 4.2) for the meeting corpus and the Fisher conversational speech corpus. Some content words like *speech*,

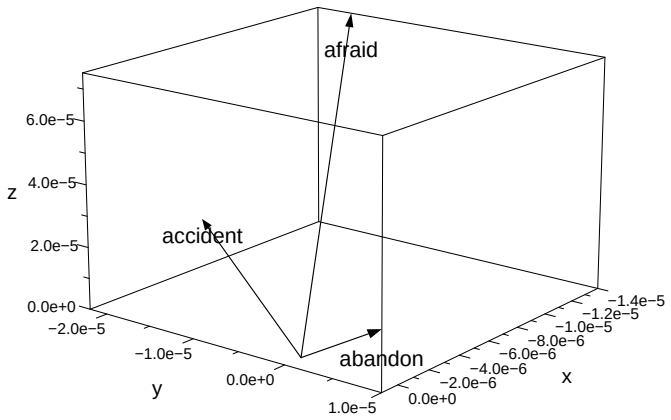


Figure 4.2: *Word vectors in the 3-dimensional space.*

numbers and *recognition* have high ϵ_{w_i} values similar to function words like *the*, *and* and *have* in the meeting corpus, because the ICSI meeting corpus that is a large part of the whole meeting corpus has a specific topic, which is "speech recognition".

w_i	Meeting ϵ_{w_i}	Fisher ϵ_{w_i}
the	0.981	0.983
and	0.979	0.982
have	0.978	0.978
recognition	0.828	0.510
speech	0.826	0.602
numbers	0.826	0.602
flower	0.186	0.414

Table 4.3: *Word un-informativeness in meetings.*

The ϵ_{w_i} values that are used as interpolation parameters for the INFG interpolation differ between one corpus and another. In the Fisher corpus these content words have a higher informativeness. One possible way to think of these values is as indicators if a corpus contains a content-word-like distribution of a word and thereby contains semantic information concerning this word. In this case the respective model should be considered in predicting the word.

Figure 4.2 shows three word vectors of the form $u_i S$ derived from broadcast news data which are drawn using the three most significant dimensions of the word vectors.

	abandon	*accident*	*afraid*
abandon	1.00	0.64	0.70
accident	0.64	1.00	0.75
afraid	0.70	0.75	1.00

Table 4.4: *Cosine similarity of word vectors.*

The similarity between the words is given by the cosine similarity of the word vectors shown in Table 4.4. These similarities are computed by using all dimensions of the LSA model. In this example *accident* is closer to *afraid* (0.75) than to *abandon* (0.64), while *afraid* is almost equally similar to *accident* (0.70) and *abandon* (0.75). These similarities can be thought of as semantic associations between the words and are used for word clustering.

4.4.2 Perplexity space of combined LSA models

Figure 4.3 shows the perplexities for the meeting and the Fisher LSA model, that are interpolated with an n-gram model using linear interpolation (Definition 4.11) tested on the meeting heldout data (Table 4.8). η_1 and η_2 are the corresponding LSA model weights. Zeros are plotted where the interpolation is not defined that is where $\eta_1+\eta_2 \geq 1$, which would mean that the n-gram model gets zero weight. The parameter of the n-gram model is $\eta_3 = 1 - (\eta_1 + \eta_2)$.

This figure shows that the minimum perplexity is reached with $\eta_1 = \eta_2 = 0$. Furthermore one can see that the graph gets very steep with higher values of η. This is beneficial for the gradient descent optimization since it is always clear where to go to reach the minimum perplexity. The minimum perplexity is however reached when the LSA model is not used at all and solely the n-gram model is used.

Figure 4.4 shows the perplexity space of the INFG interpolation (Definition 4.12) for the meeting and the Fisher model that is much flatter than the linear interpolation space. The difference in steepness can be estimated by looking at the perplexity scale, which spans $[67, 72]$ for the INFG interpolation compared to $[0, 1000]$ for the linearly interpolated models. Therefore, the parameter optimization is harder and slower for this interpolation.

On the other hand an improvement over the n-gram model can be achieved when using this interpolation. The optimum perplexity is not reached when giving both LSA models $\theta_i = 0$, but when setting the parameter for the Fisher model to $\theta_2 = 0$ and the meeting model parameter to $\theta_1 = 1$. The θ_i's have only the function of Boolean model selectors in this 2-model case. There is still the word entropy that is varying

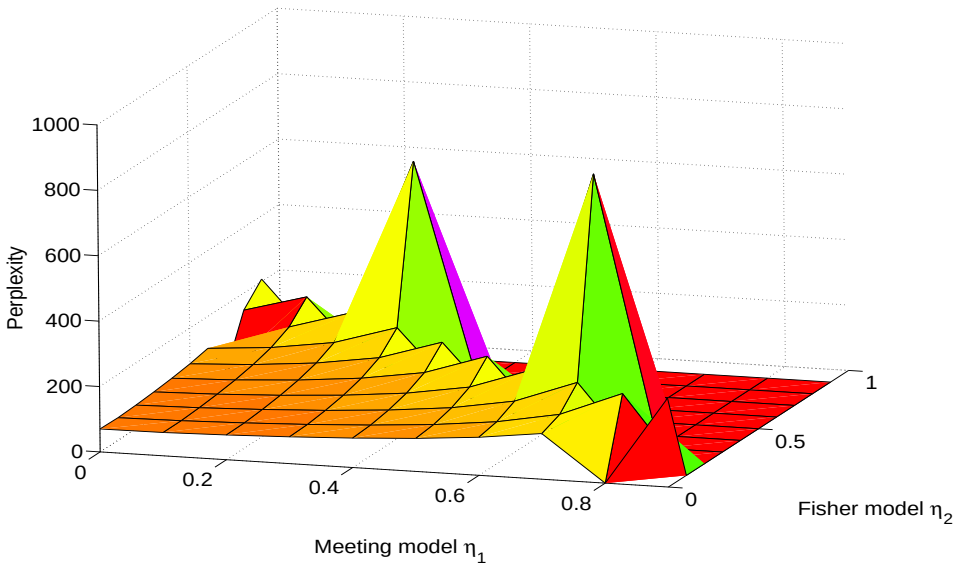


Figure 4.3: *Perplexity space for 2 linearly interpolated LSA models.*

the interpolation weight between the LSA and n-gram models.

When conducting WER experiments with combinations of more than two LSA models (Section 4.5.9), gradient-descent optimization is used to optimize all the interpolation parameters together. In this section, we have used a brute-force approach to get a picture of the whole perplexity space.

4.4.3 The repetition effect: LSA models and cache models

Some improvements of LSA-based language models over n-gram models are surely due to the redundant nature of language and speech. A lot of words that pop-up in a meeting for example are likely to pop-up again in a short window of context. A word is highly similar to a context when the word appears in the context. A cache based language model can exploit this fact by keeping a cache of words that already have been seen, and giving them higher probability (Kuhn and De Mori, 1990). To test if the performance of LSA-based models only rests on this cache-effect the word probabilities of the models are compared.

The cache or history are the words that have appeared in the text up to a certain word. In cache-based models the similarity of the word to the history/cache increases, if the word is in the history/cache (e.g. it has already been seen). In LSA-based models

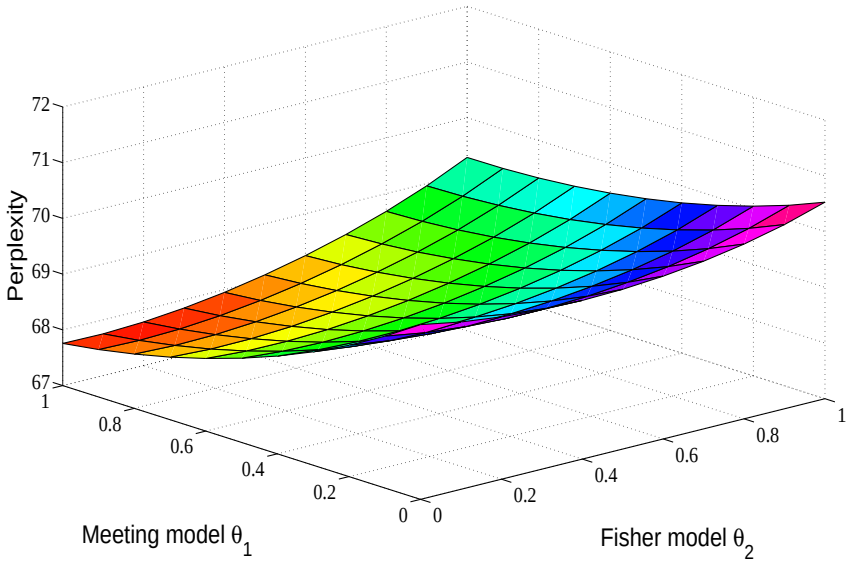


Figure 4.4: *Perplexity space for 2 INFG interpolated LSA models.*

the history is represented by a pseudo-document vector. This pseudo-document vector is constructed from the representations of the words that appear in the history/cache. The word that is to be predicted is also represented as a vector. Thereby the similarity between the history/cache and the word can be measured, even when the word does not appear in the history. To measure the cache-effect for LSA-based models one has to measure the prediction performance of the models for words that are in the history/cache, shown in Table 4.5 as 'word in hist.' and words that are not in the history/cache (Table 4.5).

Table 4.5 shows the number of improved (+) and not-improved (−) word probabilities for the meeting and the Fisher LSA model compared to a 4-gram model trained on the meeting and the Fisher data, tested on the heldout data. '+' means that the probability of the LSA model is higher than the n-gram model probability, '−' means that it is lower. The end-of-sentence event is not included.

To compute these numbers the LSA and n-gram models are applied to the test data. Then the two model probabilities are compared for each word and the number in the '+' or '−' class is increased. Finally we check if the predicted word already appeared in the history and increase the count for the 'word in hist.' or 'word not in hist.' class.

For the meeting model 60% of the improvements are due to the cache-effect where

	+ Meeting	+ Fisher	− Meeting	− Fisher
Word in hist.	54K (60%)	57K (64%)	5K (5%)	5K (6%)
Word not in hist.	6K (8%)	5K (6%)	25K (27%)	23K (24%)
90K (100%)	60K (68%)	62K (70%)	30K (32%)	28K (30%)

Table 4.5: *Number of improved (+) and not-improved (−) LSA word probabilities.*

the word appears in the history. 8% of the improvements are achieved for words that are not in the history (Table 4.5). This improvement must rely on a feature of LSA-based models that goes beyond the cache effect.

The first value is so high (60%) because the decay parameter is used in the LSA model, such that a word vector disappears from the pseudo-document. But when counting if a word has already appeared in the history/cache, all words seen so far are considered. This increases the cache-effect.[7] So a certain amount of this improvement is actually due to the semantics of the LSA model. This happens because the word vector decays in the pseudo-document but the word stays in the cache for the whole meeting. The percentage of the classes '+/Meeting/Word not in hist.' and '+/Fisher/Word not in hist.' has to be increased by this amount.

To estimate the improvements of the LSA model, that are achieved for words not already in the history, we estimate the length of the history/cache for the case when all words seen in the test data so far are added to the history/cache. This number can then be compared to the estimated number of words that are present in the pseudo-document. The first number can be estimated by assuming that each meeting contains ≈ 7500 $(90,455/12$ meetings) words. The mean length of the history for a document of length k is given by the arithmetic mean $\frac{0+1+2+\ldots+k-1}{k} = \frac{k+1}{2}$. In the above case the mean length of the history is ≈ 3700. So given a mean length of around 3700, for the meeting model 60% (Table 4.5) of the words belong to the class '+/Meeting/Word in hist.' (≈ 4500 words). But if we assume that only the last 100 words are present in the pseudo-document, some improvement of the LSA model also falls into the class '+/Meeting/Word not in hist', which must therefore be significantly higher than 8% for the meeting models (Table 4.5).

Suppose that word w_i appears at position 456 in the text ($w_{i_{456}}$), and that the meeting LSA model has a higher probability for that word than the meeting n-gram model. Assume that w_i also appears once at position 123 ($w_{i_{123}}$). This will increase the counts for '+/Meeting/Word in hist.'. But the LSA model already forgot that word,

[7] It is only a feature of our experimental setup to search among all words seen so far, and not an intrinsic property of the models.

so the appearance of the word in the history cannot be the reason for the improvement of the LSA model. Therefore the counts '+/Meeting/Word not in hist.' should be increased.

	+ Meeting		+ Fisher		− Meeting		− Fisher	
Word in hist.	164	139	140	117	1545	2399	1428	2148
Word not in hist.	31,132	21,482	45,432	28,116	36	40	87	101
	284	235	238	193	68	80	150	186

Table 4.6: *Perplexities for n-gram and LSA models.*

Table 4.6 shows the perplexities for the models. Each cell contains two values, the first for the n-gram perplexity, the second for the LSA perplexity. The small classes ('+/Meeting/Word not in hist.', '+/Fisher/Word not in hist.', '−/Meeting/Word in hist.', '+/Fisher/Word in hist.' in Table 4.5) get a very high perplexity.

For the significance experiments the probabilities of the different classes for the different models were compared. For each class (e.g. '+/Meeting/Word not in hist.') two samples were compared. One sample contains the LSA probabilities for the words in that class, the other sample contains the n-gram probabilities. The size of the samples is the same as the class sizes in Table 4.5.

According to t-tests for paired samples the differences between LSA and n-gram models for the following classes are significant ($p < 0.05$):

'+/Meeting/Word in hist'
'+/Fisher/Word in hist'
'−/Meeting/Word in hist.'
'−/Fisher/Word in hist.'
'−/Meeting/Word not in hist.'
'−/Fisher/Word not in hist.'

The differences within the classes '+/Meeting/Word not in hist.' and '+/Fisher/Word not in hist.' are however not significant, but as already mentioned the true size of this class is bigger than the estimated size.

This analysis shows that LSA-based models cannot be simply replaced by cache-based models. Although the repetition effect is important for LSA models they also cover other information.

4.4.4 The similarity exponent effect: γ exponent optimization

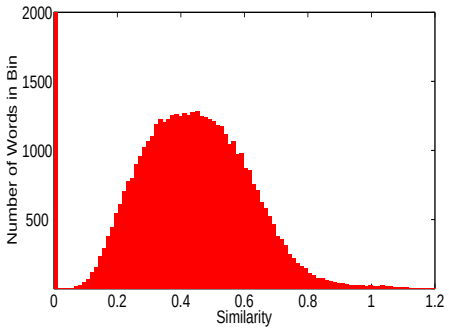


Figure 4.5: *Similarities for* $\gamma = 1$.

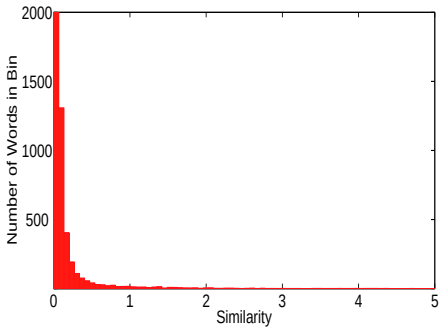


Figure 4.6: *Similarities for* $\gamma = 8$.

The parameter γ (Definition 4.9) is used to extend the small dynamic range of the LSA similarity (Coccaro and Jurafsky, 1998). Here this parameter is optimized and it is shown how it influences the LSA similarities. The similarities are scaled by using the minimum similarity given the history as in Definition 4.9. Otherwise the exponent would turn negative similarities into positive values. Figure 4.5 and 4.6 show the similarity distributions for γ values of 1 and 8, for the Fisher model on the heldout data. The second distribution expands the similarity range and concentrates a lot of similarities around zero. The high value for zero similarities in both figures comes from the zero similarities given to words not in the LSA model. Additionally some small similarities get close to zero with a higher exponent in the second figure.

All similarities < 1.0 in Figure 4.5 become smaller in Figure 4.6. This is due to the nature of the exponentiation where all values between $[0, 1]$ get smaller if exponentiated. To avoid this for similarities in the interval $[1 - \beta, 1]$ one can add an offset $\beta \in [0, 1]$ to the similarities. For $\beta = 1$ no similarities get smaller if exponentiated since all similarities are bigger than or equal to 1. Then the similarity distribution gets flatter. β is also optimized to find the effect of β values that are smaller than 1. The optimization of β is done with a brute-force approach shown for the meeting (Figure 4.8) and the Fisher (Figure 4.7) model.

For this work it is interesting to see which γ values optimize the perplexity on the heldout data. Figure 4.7 shows perplexities of the Fisher model on the heldout data for different values of γ and β.

One can see that the lowest perplexity for all β values is nearly the same while only the exponent is shifting. It can also be seen that all LSA models outperform the 4-gram model even for $\gamma = 1$. The optimal γ value for the meetings (Figure 4.8) is for all β smaller than for the Fisher model (Figure 4.7). One generalization that can be drawn from experiments with the meeting and background models in Section 4.5 is

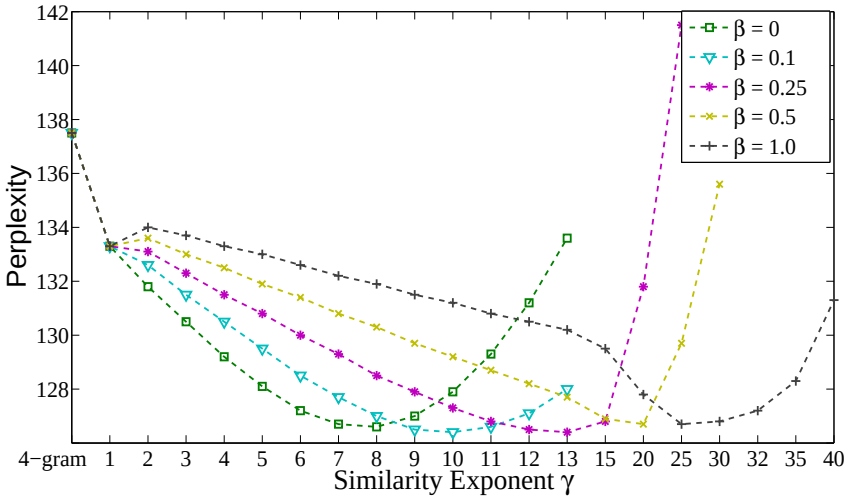


Figure 4.7: *Perplexities for the Fisher LSA model with different γ and β values.*

that the optimal γ value is in general higher for bigger models, that is models which are trained on larger corpora. This is also reflected in the relation between the meeting model and the Fisher model which can be seen from figure 4.8 and 4.7.

With the first approach (not using β or setting it to 0) one comes up with a much smaller exponent than with the second ($\beta \in (0, 1]$). It is conjectured that the different values of exponents found in the literature ranging from 7 (Coccaro and Jurafsky, 1998) to 20 (Deng and Khudanpur, 2003) are due to the usage of different values of β. Since a difference in perplexity cannot be reported there is no reason to favor one of the approaches. γ depends on the training data, so it is necessary to optimize it for each model. After the model is trained the γ values are optimized on the heldout data. Then the optimized models are used for testing.

The similarity exponent parameter is optimized independently from the interpolation parameters. It has been shown that a change in β does not affect the minimum perplexity that can be reached. Therefore β is not used in the experiments in Section 4.5. There we also found that the optimal γ value is quite stable for different test data sets, for a fixed model. Different models however do have different optimal γ values.

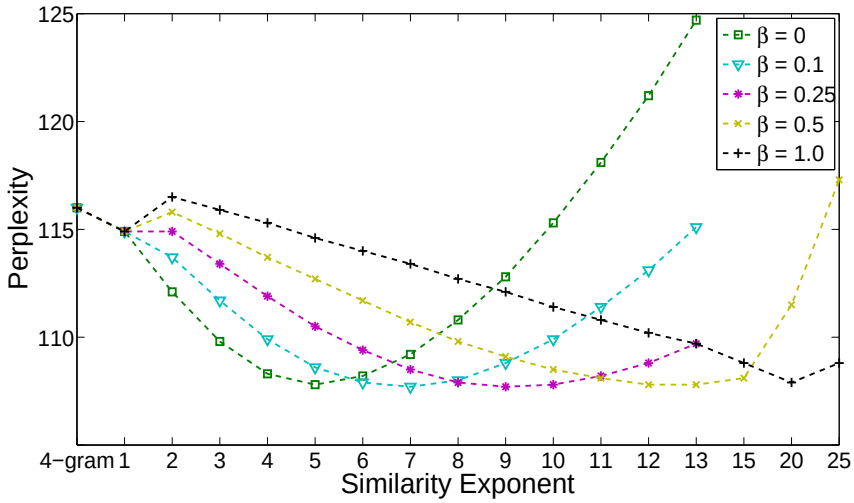


Figure 4.8: *Perplexities for the meeting LSA model with different γ and β values.*

4.4.5 The history effect: δ decay optimization

Here it is shown how the decay parameter δ (Definition 4.5) influences the perplexity. The 4-gram Fisher and meeting models are again the baselines. As a test set the meeting heldout data is used. The idea of the decay parameter is to update the pseudo-document in a way that words that were recently seen get a higher weight than words that are in a more distant history. Finally the words that are far away from the actual word are forgotten and have no more influence on the prediction of the actual word.

Figure 4.9 shows that there is a constant drop in perplexity as the length of the history increases. There is however not much difference for the decay value 0.98 and 1.0, which means that no words are forgotten. But even the shortest history with a decay of 0.05 has a lower perplexity than the 4-gram for the Fisher and the meeting model on the heldout data. It can be concluded that it is beneficial for the models not to forget too fast but there is no big difference between forgetting very slowly and never. In the experiments below, nevertheless, a decay of 0.98 is used because it still has the best performance concerning perplexity and because a similar value (0.975) was also found to be optimal by others (Bellegarda, 2000a) for the Wall Street Journal recognition task. Bellegarda (2000a) optimized the decay parameter on the WER and not on perplexity.

Since the decay parameter has similar optima for domains like meetings and broad-

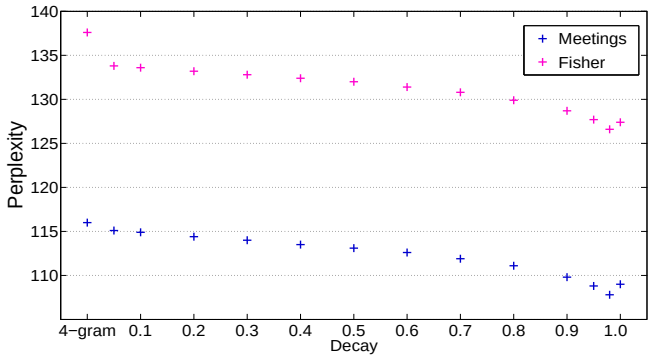


Figure 4.9: *Perplexities for a 4-gram and LSA models with different decays δ.*

cast news (Wall Street Journal) we think that this value reflects an intrinsic property of LSA-based LMs. The decay parameter is optimized independently of the interpolation parameter, and the similarity exponent parameter.

4.5 Experiments

4.5.1 Data sources

This section describes all data sources that have been used in this chapter. For the training and testing of models the data is often divided into a training data set, a development test set or heldout data set, and a test set. The training data set is used to train the models. The heldout data or development test set is used to optimize model parameters. Finally the optimized models are tested on the test data set.

Table 4.7 shows the different training data sets. There are two meeting corpora, one *Conversational Telephone Speech* (CTS) corpus, two broadcast news corpora, and two corpora of web data that are used for LM training. Since the corpora are only used for language modeling, only the transcriptions are used. The web data corpora are however pure text corpora, e.g. there is no speech data such that they would be transcriptions of it.

The web data is collected by performing a web search using n-grams from the meeting (Meet-Web) or n-grams from the Fisher corpus (FWeb). The documents that are returned by this search are then cleaned and added to the Meet-Web or FWeb corpus (Bulyko *et al.*, 2003).

Table 4.7 and 4.8 shows the number of words, type, number of documents, number of speaker turns, and a reference for the corpora. The *Linguistic Data Consortium* (LDC)

Corpus	Words ($\times 10^3$)	Type	Turns	Docs	Reference
Meetings (ICSI, NIST, CMU)	885	Meeting trans.	120,626	94	(Janin *et al.*, 2003) LDC2004T04 LDC2004T13 LDC2004T10
Fisher	19,678	CTS trans.	1,845,272	15,151	(Cieri *et al.*, 2004) LDC2004T19 LDC2005T19
Hub4-LM96	130,850	Bnews trans.	8,570,146	125,411	LDC98T31
TDT4	11,869	Bnews trans.	479,569	608	LDC2005T16
Meet-Web	147,510	Web data	11,435,875	132,105	(Bulyko *et al.*, 2003)
FWeb	530,284	Web data	45,606,777	147,480	(Bulyko *et al.*, 2003)

Table 4.7: *Training data sources.*

number is included in the reference if the corpus is available from LDC. What is considered as a document and a speaker turn varies depending on the corpus. For the meeting corpora the documents are meetings, for the CTS corpora they are conversations, for the broadcast news corpora they are broadcast news stories, and for the web data they are web documents. For all corpora the speaker turns are regular speaker turns, except for the web data where they are written sentences. The meeting corpus includes 94 meetings, 11 NIST meetings, 14 CMU meetings, and 69 ICSI meetings.

Table 4.8 shows the different test data sets. As development test data the heldout data set is used. This data set consists of 12 meetings. 4 ICSI, 4 CMU, and 4 NIST meetings taken from the original meeting corpus. The meeting corpus that is used to train the meeting models ('Meetings (ICSI, NIST. CMU)' in Table 4.7) is the original meeting corpus minus the heldout meetings.

As test data sets the sets RT02-DEV, RT04-S-DEV and RT05-S were used. RT02-DEV is a development test set from the NIST Rich Transcription 2002 (RT-02) meeting evaluation which consists of two ICSI meetings. RT04-S-DEV is the development test set from the NIST Rich Transcription 2004 Spring (RT-04S) meeting evaluation. RT05-S is conference room meeting test data from the NIST Rich Transcription 2005 Spring (RT-05S) meeting evaluation (Fiscus *et al.*, 2005). Since this test data is used for perplexity and WER experiments it contains meeting transcriptions and speech

Corpus	Words ($\times 10^3$)	Type	Turns	Docs	Reference
RT02-DEV	37	Meeting trans.	5957	2	
RT04-S-DEV	17	Meeting trans.	1680	8	
Meeting heldout data	90	Meeting trans.	12,876	12	(Janin *et al.*, 2003) LDC2004T04 (Garofolo *et al.*, 2004) LDC2004T13 (Burger *et al.*, 2004) LDC2004T10
RT05-S	23	Meeting trans. & speech	3340	10	(Fiscus *et al.*, 2005)

Table 4.8: *Test data sources.*

data. The speech data is used in the decoding process.

For the perplexity experiments mostly the RT04-S-DEV data was used. For the optimization and analysis of the models the meeting heldout data was used. For the WER experiments RT05-S test data was used. The RT05-S test data was not available when the perplexity experiments were conducted.

4.5.2 Perplexities for meeting models

For the training of the first models the ICSI meeting corpus (Janin *et al.*, 2003) is used. For the test the 2002 meeting evaluation development set (RT02-DEV) (Table 4.8) is used. Meeting boundaries are taken as document boundaries, which are needed for the training of the LSA model.

Table 4.9 shows the perplexity results for ICSI meetings for the different methods. As a development test set for optimization the ICSI part of the heldout data set (Table 4.8) was used. The γ parameters were optimized using gradient descent. For the meeting models a value of 5 was found to be optimal. While the "linear interpolation" (LIN) and the "similarity modulated n-gram" (SIMMOD) do not bring any improvements over the baseline 3-gram model, the "information weighted geometric mean" (INFG) reduces perplexity. The improvement of the "information weighted geometric mean" interpolation over the 3-gram model is consistent with findings in Deng and Khudanpur (2003). For the other interpolations always the INFG method is applied since it outperformed all other interpolation methods.

Model	Perplexity
3-gram	84.3
INFG	81.7
SIMMOD	85.1
LIN	88.2

Table 4.9: *Perplexity results for ICSI meetings on RT02-DEV.*

The next meeting model is trained on CMU, ICSI, and NIST meetings (Table 4.7). For these and the other tests, the NIST Rich Transcription 2004 Spring (RT04-S) meeting evaluation development test set (RT04-S-DEV in Table 4.8) is taken. The optimization was done on the heldout data set (Table 4.8). Table 4.10 shows the perplexities for this model interpolated using the INFG method with the n-gram model that is estimated with modified Kneser-Ney smoothing (Chen and Goodman, 1998). There are small improvements over all meetings.

According to t-tests for paired samples the differences between LSA models and n-gram models are significant for the ICSI meetings ($p < 0.001, N = 4898$) and for all meetings ($p < 0.001, N = 18853$). No significant differences are found for the CMU ($p = 0.12, N = 4235$), LDC ($p = 0.61, N = 5225$), and NIST ($p = 0.11, N = 4492$) meetings. The reason for the significant improvement on the ICSI meetings is that the ICSI data make up most of the meeting training data.

Model	All	CMU	ICSI	LDC	NIST
4-gram	129.9	176.4	77.1	160.1	134.7
INFG	125.4	170.1	75.9	152.1	129.7

Table 4.10: *Perplexity results for all meetings on RT04-S-DEV.*

4.5.3 Perplexities for meeting models with topic boundaries

For the ICSI meeting corpus there is also a version that includes topic boundaries. Topic boundaries should be beneficial for LSA models since they increase the number of training documents and segment the corpus into semantically significant units. The topics are structured into topic, sub-topic and sub-sub-topic.

For these experiments three different LSA models are trained. The first is trained with all the non-ICSI meetings together with the ICSI meetings with simple document boundaries (Topic1). The second is trained with ICSI meeting training data that is structured into topics and then sub-topics (Topic2). The third used topic, sub-topic

Model	CMU	ICSI	LDC	NIST
4-gram	172.9	76.4	156.3	132.2
Meeting-LSA	168.5	75.4	149.0	127.7
Topic1	169.3	77.7	151.2	130.8
Topic2	171.7	77.9	153.6	132.0
Topic3	171.1	78.4	153.1	132.4

Table 4.11: *Perplexity results for meetings with topic boundaries on RT04-S-DEV.*

and sub-sub-topic (Topic3). This results in a document number of 523 for the Topic1-corpus, 1293 for the Topic2-corpus, and 1516 for the Topic3-corpus in comparison to 106 documents (Meeting data and meeting heldout data) in the original meeting corpus.

The topic boundaries are given as part of a structured *eXtensible Markup Language* (XML) document. To expand this representation into a flat representation containing chunks of text separated by document boundaries, the text belonging to a sub-topic is included in the corresponding topic text, and the text belonging to a sub-sub-topic is included in the corresponding sub-topic and topic text. In this way the fine-grained text is repeated. To avoid repetition one could count a sub-topic text not as part of the topic text, which would make the topic text possibly very small.

As Table 4.11 shows there is no perplexity gain for the topic models over the meeting-LSA model. Since topic boundaries are available for the ICSI meetings, at least a gain in perplexity for the ICSI data was expected when using the topic models. But as can be seen from Table 4.11 the more fine-grained the topic structure is, the worse the perplexity. The reason for this can be that the documents with the fine-grained topic boundaries (Topic2, Topic3) get too small. Another reason for the poorer performance could be that there are some topics like “introduction” and “end” that are more like agenda-items than topics.

4.5.4 Perplexities for background domain models

Since the training corpora for meetings are very small further LSA models are trained on multiple background domains. A mixture of language models trained on adaptation and background domains has also been used for word-based n-gram models for meetings by Stolcke *et al.* (2005).

The transcripts of the following widely-used corpora are used: Fisher, Hub4-LM96 and TDT4 (see Table 4.7). Furthermore data collected from the web, similar to CMU, ICSI and NIST meetings (=Meet-Web), and the Fisher corpus (=FWeb) is used. All

data sources are shown in Table 4.7. All text corpora together contain almost 1 billion (837 million) word tokens. For practical reasons an n-gram mixture model is trained on each corpus separately and then these corpus models are interpolated to yield one model. The same strategy is chosen for training an LSA model on all available data.

The documents for the Fisher data are conversations, for the Hub4-LM96 broadcast news data they are news stories, and for the web data websites are taken as documents. The word-based n-gram model (Stolcke *et al.*, 2005) used in ASR for multi-party meetings in the 2005 meeting recognition evaluation (Fiscus *et al.*, 2005) is also trained on the same data. When first interpolating meeting LSA models with this large meeting n-gram model no improvement was seen. This finding motivated us to include data from background domains.

Model	Hub4-LM96	Tdt4	Meet-Web	Fisher
4-gram	144.1	238.8	145.5	131.5
INFG	132.0	224.6	134.6	121.2

Model	FWeba	FWebb	FWebc	FWebd
4-gram	130.0	130.0	130.0	130.0
INFG	120.8	119.3	119.2	119.9

Table 4.12: *Perplexity results for background domain models on RT04-S-DEV.*

Table 4.12 shows the estimated perplexities for the models trained on the background domain data. The corresponding n-gram models are trained on the same data. The Fisher web data FWeb is divided into four parts FWeba, FWebb, FWebc, and FWebd because it is too big to train one LSA model on it. The test set is again the RT04-S-DEV test set. There are improvements over all background domains.

According to t-tests for paired samples all differences between n-gram models and LSA models in Table 4.12 are significant ($p < 0.001, N = 18853$). These results are promising concerning the interpolation of multiple LSA models. Each LSA model can improve the n-gram model.

Concerning the γ exponent parameter defined in (4.9) that is used to expand the small dynamic range of the LSA similarity, it is found that the optimal value of γ is higher for bigger models. The optimal value for the meeting model is 5, for the Fisher model it is 7 and for all other models it is 9, using an offset $\beta = 0$.

4.5.5 Perplexities for combined LSA models

Figure 4.10 shows optimized θ_i values for the INFG interpolation (Definition 4.12) of meeting and background LSA models. The meeting model gets most of the weight, followed by the Fisher and FWeb data. The meeting model also gets most of the weight for the n-gram mixture models. Since the Fisher data is conversational speech

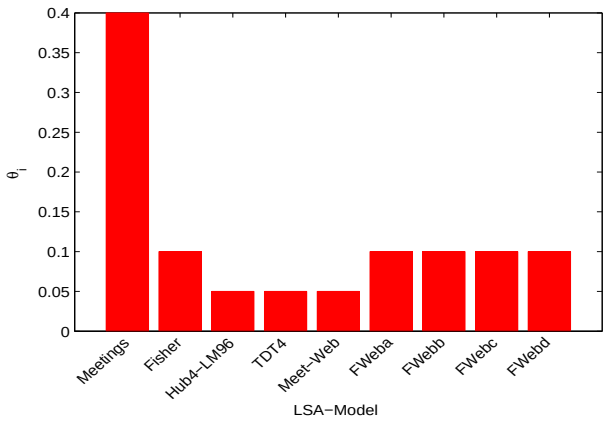


Figure 4.10: *Optimized θ_i values for INFG interpolation of LSA models.*

data, and the FWeb data is web data collected according to the Fisher corpus, the weights for these models are justified. It is interesting that the Meet-Web data gets a lower weight, although it is web data collected on the basis of the meeting data.

Model	All	CMU	ICSI	LDC	NIST
4-gram mixture model	85.4	104.1	67.0	87.5	89.8
LIN	85.4	104.1	67.0	87.5	89.8
INFG	84.5	103.0	66.2	86.4	88.9

Table 4.13: *Perplexity results for combined LSA models on RT04-S-DEV.*

Perplexity results for the combination of all of the eight background LSA models, the meeting LSA model, and the n-gram mixture model trained on all the available data are given in Table 4.13.

In case of the linear interpolation all LSA models get zero weight, so there is no improvement over the n-gram model. The INFG interpolation gives a small improvement, where the highest θ_i weights are given to the meeting LSA model, followed by the models trained on the Fisher and the Fweb data. According to t-tests for paired samples all differences between n-gram models and LSA models in Table 4.13 are significant ($p < 0.001$).

4.5.6 Word-error-rate (WER) for meeting models

For the WER experiments, conference room meeting test data from the NIST Rich Transcription 2005 Spring (RT-05S) meeting evaluation (Fiscus *et al.*, 2005) is used, which contains meetings from several different sites. During decoding three language models are used. A bigram is used for word lattice generation, a trigram is used for decoding from word lattices, and a 4-gram is used for N-best list rescoring (Stolcke *et al.*, 2005). After that the LSA model trained on the same data is used to rescore the N-best lists again and the results are compared to the 4-gram baseline. The number N varies depending how many hypotheses are generated in the decoding process.

It can happen that the elements of N-best lists are very short or that they are only chunks of sentences. In this case the full potential of LSA-based models, which lies in the modeling of the long-term history cannot be realized. The n-gram language model for example is applied to the N-best list elements without taking previous N-best lists into account. To realize the potential of LSA models, the N-best lists are ordered, and the first-best element of the previous N-best list is added to the pseudo-document. In this way the long-term history is encoded. The drawback of this approach is that the first-best element need not be the correct one. But as the LSA model performance is dependent on wider contexts (Section 4.4.5) this approach is justified.

	n-gram	LSA	Relative change
AMI	25.5	25.5	$\mp 0.0\%$
CMU	24.7	24.9	-0.8%
ICSI	19.8	19.5	$+1.5\%$
NIST	25.7	25.8	-0.3%
VT	27.0	26.9	$+0.3\%$
ALL	24.9	24.8	$+0.4\%$

Table 4.14: *Word-Error-Rates in % for Meeting LSA models.*

The relative WER improvements on the meeting data are neither significant for the ICSI data ($+1.5\%$, $p = 0.114$), nor for the *Virginia Tech* (VT) data ($+0.3\%$, $p = 0.393$) or for the complete data set ($+0.4\%$, $p = 0.791$) according to a matched pairs test (Hunt, 1988; Gillick and Cox, 1989).

When rescoring N-best lists it is however only possible to improve the recognition rate if the correct result, or a result that is closer to the correct result than the actual 1-best element is present in the list. If no better list entry is present the WER can only increase or stay the same for this list.

4.5.7 WER for meeting models with topic boundaries

Since no improvements in terms of perplexity have been achieved by using the topic models instead of the regular meeting models, topic models are not used for rescoring.

4.5.8 WER for background domain models

	n-gram	LSA	Rel. change
AMI	24.9	25.1	−0.8%
CMU	26.0	26.2	−0.7%
ICSI	23.9	23.9	∓0.0%
NIST	25.5	25.7	−0.7%
VT	25.0	25.1	−0.4%
ALL	25.1	25.2	−0.3%

Table 4.15: *WER in % for Hub4-LM96.*

	n-gram	LSA	Rel. change
AMI	28.8	29.1	−1.0%
CMU	28.4	28.7	−1.0%
ICSI	25.7	26.2	−1.9%
NIST	28.4	28.7	−1.0%
VT	26.6	26.7	−0.3%
ALL	27.5	27.8	−1.0%

Table 4.16: *WER in % for Tdt4.*

	n-gram	LSA	Rel. change
AMI	26.4	26.6	−0.7%
CMU	27.3	28.1	−2.9%
ICSI	24.1	24.5	−1.6%
NIST	25.9	26.0	−0.3%
VT	25.8	25.8	∓0.0%
ALL	25.9	26.2	−1.1%

Table 4.17: *WER in % for Meet-Web.*

	n-gram	LSA	Rel. change
AMI	24.1	24.2	−0.4%
CMU	25.9	25.6	+1.1%
ICSI	23.8	24.0	−0.8%
NIST	25.0	24.8	+0.8%
VT	24.6	24.8	−0.8%
ALL	24.7	24.7	∓0.0%

Table 4.18: *WER in % for Fisher.*

	n-gram	Weba	Rel. change Weba / n-gram	Webb	Rel. change Webb / n-gram
AMI	26.0	25.8	+0.7%	25.9	+0.3%
CMU	26.5	26.5	∓0.0%	26.4	+0.3%
ICSI	23.5	23.1	+1.7%	23.2	+1.2%
NIST	25.6	25.7	−0.3%	25.6	∓0.0%
VT	24.0	24.9	−3.7%	24.5	−2.0%
ALL	25.2	25.3	−0.3%	25.2	∓0.0%

Table 4.19: *WER in % for Fisher-Weba and Webb.*

For the background domain models significant WER improvements are achieved for two of the data and meeting sites according to a matched pairs test. For the Fisher data a relative WER improvement of +1.1% on the CMU data is achieved that is

	n-gram	Webc	Rel. change Webc / n-gram	Webd	Rel. change Webd / n-gram
AMI	26.0	25.9	+0.3%	26.0	∓0.0%
CMU	26.5	26.6	−0.3%	26.6	−0.3%
ICSI	23.5	23.1	+1.7%	23.3	+0.8%
NIST	25.6	25.3	+1.1%	25.7	−0.3%
VT	24.0	24.6	−2.5%	24.3	−1.2%
ALL	25.2	25.2	∓0.0%	25.3	−0.3%

Table 4.20: *WER in % for Fisher-Webc and Webd.*

significant ($p = 0.052$). For the FWebc data a relative WER improvement of +1.7% on the ICSI data is achieved that is also significant ($p = 0.043$).

For the other background domain models that improved the WER the improvements are not significant according to a matched pairs test. These are the +0.8% improvement of the Fisher model on the NIST data ($p = 0.235$), the +0.7% and +1.7% of the FWeba model on the AMI ($p = 0.698$) and ICSI data ($p = 0.079$), the +0.3%, +0.3% and +1.2% of the FWebb model on the AMI ($p = 0.966$), CMU ($p = 0.266$) and ICSI ($p = 0.204$) data, the +0.3% and +1.1% of the FWebc model on the AMI ($p = 0.612$) and NIST data ($p = 0.156$), and the +0.8% of the FWebd model on the ICSI ($p = 0.259$) data.

4.5.9 WER for combined LSA models

	n-gram	LIN	Rel. change LIN / n-gram	INFG	Rel. change INFG / n-gram
AMI	24.7	24.7	∓0.0%	24.5	+0.8%
CMU	26.5	26.5	∓0.0%	26.7	−0.7%
ICSI	22.6	22.6	∓0.0%	22.7	−0.4%
NIST	24.4	24.4	∓0.0%	24.4	∓0.0%
VT	24.4	24.4	∓0.0%	24.4	∓0.0%
ALL	24.5	24.5	∓0.0%	24.6	−0.4%

Table 4.21: *Word-Error-Rates in % for Combined LSA models.*

For the combination of LSA models using INFG interpolation (Table 4.21) a relative WER improvement on the AMI data of +0.8% is achieved, which is not significant ($p = 0.236$) according to a matched pairs test.

4.6 Summary and discussion

After discussing other studies on the semantics of LSA-based models, dealing with semantic concepts from different levels, we show that LSA models can cover the concept of synonymy to a certain extent. Then the mathematical background of LSA-based modeling is introduced.

We show how to optimize the parameters for interpolated LSA-based language models and show that simple linear interpolation does not achieve any improvements. With the INFG interpolation an improvement is achieved.

The comparison between LSA and cache-based models shows that a large amount of the improvement is due to the repetition of words, but there is also an improvement that relies on other features of the LSA-based models. So cache-based models cannot simply replace LSA-based models.

The optimization of similarity exponent and offset is presented and the relation between offset selection and similarity exponent is investigated. The optimization of the decay parameter shows that it makes little difference if the LSA model never forgets or forgets very slowly, but a bigger difference if the model forgets fast.

From the analysis and optimization of LSA-based language models we can conclude that parameter optimization is crucial for outperforming word-based n-gram language models by LSA models.

When first interpolating the meeting LSA model with a mixture n-gram model that is trained on all available data (Table 4.7) no improvements in terms of perplexity is seen. Our conclusion is that the LSA model does not capture more information than the n-gram model in this case.

But when training on the same background and meeting data an improvement of the LSA model over the n-gram model in terms of perplexity is achieved. This shows that the LSA models capture some additional information compared to the n-gram model, if trained on the same data. Our analysis also shows that the predictive power of the models is partly due to the repetition of words, which occurs frequently in conversational speech.

Since it is not feasible to train one LSA model on all the data the next step is to think about combinations of LSA models. The most promising technique is the log-linear INFG interpolation with optimized interpolation weights. A generalized version of the INFG interpolation method for multiple LSA models is introduced. This generalization is more flexible in allowing the weighting of the global n-gram influence.

But even with this interpolation only small improvements in terms of perplexity are achieved. Concerning perplexity it can be concluded that improvements over all background domains can be achieved, but that an effective interpolation method for multiple LSA models is still missing.

One possible problem in combining multiple LSA models could be that there are too many cases where the models give different similarities to the same words and

contexts. In this case the word entropy could still make a difference with the INFG method. But with a similar entropy the models would neutralize each other.

Concerning WER significant improvements for some of the background models are achieved, but no improvements for the combined LSA models and the meeting model.

The significantly different results in terms of WER for different training corpora and meeting sites suggest that the LSA model combination should be preceded by an LSA-based matching between meeting sites and training data. A part of this matching could be a comparison of the semantic spaces of LSA models providing a semantic similarity metric for comparing two LSA models. In this way models can be combined according to a clustering of this semantic hyperspace.

In this thesis a new method is introduced for combining multiple LSA models (Definition 4.12). The advantages of this combination are that models can be trained on large amounts of data, word dependent and model dependent interpolation parameters can be used, including a model parameter for the n-gram model. The word dependent parameters are different for each model, such that information about a word contained in a model is considered. Although the combination only gives a small improvement in perplexity compared to an interpolated n-gram model, we think that it can be useful for other data sets or similar domains.

Furthermore LSA-based models are for the first time applied to the meeting domain. In this domain it is often necessary to combine domain models with background domain models because only small amounts of domain training data are available (Section 4.5).

5 WordNet-based semantic relatedness measures in automatic speech recognition (ASR) for multi-party meetings

> A curious thing about the ontological problem is its simplicity. It can be put in three Anglo-Saxon monosyllables: 'What is there?' It can be answered, moreover, in a word - 'Everything' - and everyone will accept this answer as true (Quine, 1953a).

5.1 Introduction

In this chapter[1] the performance of eight WordNet-based semantic similarity/distance measures[2] for word prediction in conversational speech is evaluated. It is shown that the WordNet-based models cannot be easily transformed to conditional statistical models, since WordNet models are based on a (local) concept of semantic relatedness that is based on the relatedness of two words. Nonetheless these models can be combined with standard word-based n-gram models to rescore N-best lists.

In language modeling the concept of perplexity is used to measure the performance of models. Since this concept is based on a probability distribution, a different evaluation metric is used in this chapter. First the performance of different WordNet-based relatedness measures is evaluated, then the *Word-Error-Rate* (WER) on meeting data is computed for the best performing measures.

This investigation starts with conversational telephone speech data, and then switches to multi-party meetings. Since the second type of data is similar to the first type, this thesis assumes that the performance results from 2-party conversational speech can be transferred to multi-party meetings. For the evaluation of the relatedness measures conversational telephone speech data is used. The best performing measures are applied to *Automatic Speech Recognition* (ASR) for multi-party meetings.

A ranking of the different WordNet measures is given, which shows that the performance of the measures differs significantly for noun and verb prediction. Varying dialog contexts and cross part-of-speech comparison are used.

Text-based semantic relatedness measures can improve word prediction on simulated speech recognition hypotheses as Demetriou *et al.* (2000, 1997) has shown. Demetriou *et al.* (2000) generated N-best lists from phoneme confusion data acquired from a

[1] Parts of the content of this chapter were first published in Pucher (2005).

[2] The concept "relatedness measure" subsumes "similarity measure" and "distance measure".

speech recognizer, and a pronunciation lexicon. Then sentence hypotheses of varying WERs were generated based on sentences from different genres from the *British National Corpus* (BNC). It was shown by them that the semantic model can improve recognition, where the amount of improvement varies with context length and sentence length. Thereby it was shown that these models can make use of long-term information, similar to *Latent Semantic Analysis* (LSA) models. This thesis assumes that these improvements can also be achieved with N-best lists acquired from a real speech recognition system, using such models. The additional difficulty that is faced in this work is the application of these models to conversational speech data and meeting data.

5.1.1 Word prediction by semantic similarity

The standard n-gram approach in language modeling for speech recognition cannot cope with long-term dependencies. Therefore Bellegarda (2000b) proposed combining n-gram language models, which are effective for predicting local dependencies, with LSA-based models for covering long-term dependencies. WordNet-based semantic relatedness measures can be used for word prediction using long-term dependencies, as in these examples from the CallHome English telephone speech corpus:

(5.1) B: I I well, you should see what the ⌊students⌋
B: after they torture them for six ⌊years⌋ in middle ⌊school⌋ and high ⌊school⌋ they don't want to do anything in ⌊college⌋ particular.

The notation in this chapter follows Lyons (1995, p. 24) and uses single quotation marks for words ('student'), italics for word forms (*students*), and double quotation marks for meanings ("student"). If referring to a specific meaning of a word subscripts are used ("student$_1$"). If referring to one meaning of a word, that can but does not have to be determined by the context, the subscript i is used ("student$_i$"). The terms 'meaning', 'concept' and 'sense' are interchangeable. c_i is used to refer to concepts. The notation ⌊ ⌋ is used to refer to the content words (nouns and verbs) in the context that can be used for a prediction of the last bracketed word.

In Example 5.1 *college* can be predicted from the noun context using semantic relatedness measures, here between *students* and *college*. A 3-gram model gives a ranking of *college* in the context of *anything in*. An 8-gram predicts *college* from *they don't want to do anything in*, but the strongest predictor is *students*.

(5.2) B: everyone who's who's extra busy, of course, you know who's ⌊doing⌋ the ⌊cooking⌋, like tonight it was Benny and me.
A: mm.
B: I ⌊mean⌋ e- so all the ⌊people⌋ who are ⌊working⌋.

In Example (5.2) *working* can be predicted from *people*, *cooking* and *doing*, since for verb prediction the verbs and nouns in the context are used.

In addition to such predictions based on semantic relatedness there is another type of prediction, which relies on WordNet's morphological analyzer. In these predictions a word is predicted if the word itself or an inflection of it occurs in the context.

5.1.2 Evaluation method and data

Many different similarity/distance measures for WordNet have been proposed. Here the performance of these measures for word prediction in conversational speech is evaluated. These measures are used for N-best list (Table 1.3) speech recognition hypothesis rescoring.

For the performance evaluation two context measures are defined. The first measure defines the relatedness of a word and a context (Definition 5.12) The second measure defines the relatedness between a word, an utterance, and a context (Definition 5.13), where the whole utterance can be used as an additional context for measuring the relatedness. For the sake of simplicity the term utterance is used here. In the first experiments the bag-of-words in a dialog turn is added to the context. In the second experiments N-best lists are used, whose entries are sometimes utterance-fragments or utterance-chunks.

To evaluate the performance of the WordNet measures five randomly selected dialogs from the CallHome English corpus are used. The corpus is automatically tagged using a 3-gram tagger and the Brown Corpus, and the content words (nouns, verbs) are extracted. No post-corrections are made. This results in 1316 word tokens and 1271 word types, including 685 nouns and 586 verbs. We are aware that this is a small test set compared to the meeting test sets (Table 4.8) that contain between $17,000$ and $90,000$ word tokens, but there are several reasons why we think that the results are still generally valid. First the dialogs are chosen randomly from an evaluation test set. Second the best performing measures are stable over a number of different conditions concerning our performance metric for word prediction (Section 5.3.1).

Most relatedness measures do not work across different parts-of-speech, such that one cannot compute the relatedness between a verb and a noun directly. So cross part-of-speech comparison is not used in the first run. WordNet has four parts-of-speech - nouns, verbs, adjectives and adverbs (Fellbaum, 1998, p. 9). This is already simplified in comparison to the output of most *Part-of-Speech* (POS) taggers. In the first experiments only nouns and verbs are used, to be able to compare all measures, because most measures do not work for adjectives and adverbs.

The aim of this work is to apply the results to multi-party dialogs. Therefore the whole dialog context (two speakers), as well as the sub-dialog contexts (one speaker) which are only the monologues, are used. The term 'monologue' is used for speech produced by one speaker, 'dialog' is used for two speakers. This usage does not reflect

the intuitive sense of the terms and is only introduced to differentiate the contexts.

The best performing measures are used for N-best list rescoring. The data used for rescoring is different from the evaluation data. For rescoring a wider context can be used than for word prediction, since the performance metric for word prediction requires an ordering of the whole vocabulary given a context. For the WER experiments N-best lists generated from the decoding of conference room meeting test data of the NIST Rich Transcription 2005 Spring (RT-05S) meeting evaluation (Table 4.8) are used. For these experiments nouns, verbs, and adjectives are used, since the best performing measures allow for this extension.

5.1.3 WordNet semantics

WordNet is organized into synonym sets that are related to each other by different relations. One important relation is "hypernymy" the semantic relation of subsumption. For two noun senses c_i and c_j it can be defined by the universally quantified sentence '$\Box(\forall(x)(c_i(x) \rightarrow c_j(x)))$' read as "It is necessary that, if something is a c_i then it is a c_j". '$\Box$' is a sentence operator, which means that the sentence is necessarily true e.g. true in all possible worlds. If the necessity can be established through semantic analysis, the above sentence is a meaning postulate or analytic truth (Carnap, 1956, p. 10), simply true through the meaning of its components.[3] If this sentence is true then c_j is a hypernym of c_i. Normally the term 'hypernymy' is used for an asymmetrical relation, such that '$\Box(\forall(x)(c_j(x) \rightarrow c_i(x)))$' also has to be false for c_j to be a hypernym of c_i. Otherwise the two terms would be descriptive synonyms. Lyons (1995, p.63) distinguishes "descriptive synonyms" and "expressive synonyms". While "descriptive synonymity" requires that two concepts can be substituted for each other in all contexts, the notion of synonymy and synset in WordNet is a bit more open. It only requires that the concepts are interchangeable in some contexts (Miller, 1998).

5.2 WordNet-based semantic relatedness measures

5.2.1 Definition of the measures

Eight similarity/distance measures from the Perl package WordNet-Similarity written by Pedersen *et al.* (2004) are used. The measures are named after their respective authors. All measures are implemented as similarity measures. RES (Resnik, 1995), LIN (Lin, 1997, 1998) and JCN (Jiang and Conrath, 1997) are based on the information content, the measures LCH (Leacock and Chodorow, 1998), WUP (Wu and Palmer, 1994) and PATH use path lengths between two words in the WordNet graph, and

[3] There are interesting philosophical discussions on the concepts of "analyticity", "synonymity", and "necessity" (Quine, 1953b)(Kripke, 1980, p. 34) that provide a detailed analysis of these concepts.

HSO (Hirst and St-Onge, 1998) and LESK (Banerjee and Pedersen, 2003) allow for comparison across POS boundaries.

For the definition of the first three measures one needs to define the concept of *Information Content* (IC) and LCS, where

> *...the LCS of concepts A and B is the most specific concept that is an ancestor of both A and B.* (Pedersen *et al.*, 2004)

In Figure 5.1 common subsumers of the concept "person$_i$" and "supernatural$_i$" are "causal agent$_i$" and "entity$_i$". In this case "causal agent$_i$" is the only least common subsumer between the two concepts. If there are multiple least common subsumers one has to choose among them by taking the one with the highest IC for example.

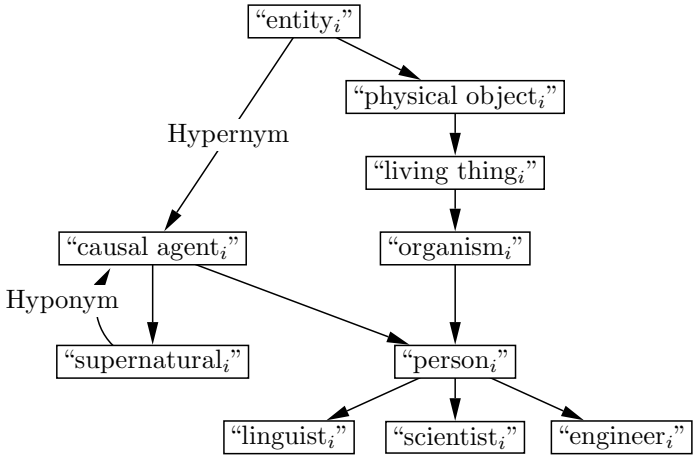


Figure 5.1: *Subgraph of WordNet.*

The count of a concept c_i, count(c_i) is the number of occurrences of the concept in a corpus. The concept frequency is defined as

Definition 5.1 *Concept frequency*

$$\text{freq}(c_i) = \text{count}(c_i) + \sum_{c_j \in \text{sub}(c_i)} \text{freq}(c_j) \ .$$

sub(c_i) are the concepts that are subsumed by c_i, that c_i is a hypernym or superordinate of. The probability of a concept c_i is given by the frequency of c_i in the corpus freq(c_i) divided by the total number of concepts in the corpus. The total number of concepts N is the frequency freq(c_k) of the top level concept c_k. When a concept is encountered in the corpus its count is also added to the frequency of the more general concepts. This is achieved by the recursive Definition 5.1.

The information content of a concept $\mathrm{IC}(c_i)$ is defined as the negative log of the probability of encountering an instance of the concept (Resnik, 1995). The more specific a concept, the higher its information content.

Definition 5.2 *Information content*

$$\mathrm{IC}(c_i) = -\log_2(\frac{\mathrm{freq}(c_i)}{N})$$

Referring to a concept here, means referring to a sense in WordNet, which is the sense of a word with a certain POS. WordNet is organized into sets of synonyms. All words have a POS and most have multiple senses. Let us assume that the ontology in Figure 5.1 can be mapped one-to-one onto the corresponding words, so forget for a moment that each word has multiple senses and a POS. Let us further assume that the 2-gram *causal agent* is an instance of the noun concept "causal agent", and the 1-gram *engineer* is an instance of the noun concept "engineer" (The 1-gram *engineer* could also be an instance of the verb concept "to engineer"). Then the following 1-gram counts and concept counts for the concepts in Figure 5.1 can be derived from parts of the Hub4-LM96 broadcast news corpus. The counts of a more specific concept like "person" are added to the counts of a more general concept like "organism".

concept	n-gram count(concept)	freq(concept)	IC
"entity"	379	218,271	0.00
"causal agent"	1	24,193	0.11
"living thing"	19	48,420	0.22
"physical object"	1	96,822	0.35
"organism"	129	24,265	0.65
"person"	20,922	22,529	0.98
"scientist"	859	859	2.40
"engineer"	732	732	2.47
"supernatural"	56	56	3.59
"linguist"	16	16	4.13
		N =218,271	

Table 5.1: *Counting concepts.*

When deriving the IC for the whole WordNet, morphological analysis should be included, such that *engineer* and *engineers* are counted as instances of the concept "engineer". POS-tagging should also be included, such that only occurrences of the noun 'engineer' are counted if encountering the noun form. This can be done with high reliability since morphological analysis and POS tagging at the level where only types of content words have to be distinguished are reliable.

In most implementations, including the one used here, there is no word-sense disambiguation applied before counting concept frequencies. An occurrence of the noun 'engineer' adds a count to all the senses of "engineer". The noun 'engineer' has the following senses in WordNet:

(5.3)
1. engineer, applied scientist, technologist – a person who uses scientific knowledge to solve practical problems
2. engineer, locomotive engineer, railroad engineer, engine driver – the operator of a railway locomotive

In principle one would have to decide which sense a word in the corpus has, and then just increment the count for this sense. Otherwise the direct counts count(c_i) are false. The errors made with the direct counts can be corrected to a certain extent because the senses are located on different places in the WordNet hierarchy. The different senses of 'engineer' have different subordinate senses, so they indirectly get different counts from the corpus through their subordinate senses. Now the prerequisites are given to define the first three measures. The measures are defined on concepts, that is WordNet senses, and not on words or word forms.

IC-based measures

Because WordNet allows multiple inheritance RES (Resnik, 1995) takes the LCS with the highest information content. IC(c_i) is the information content of c_i and LCS(c_i, c_j) are the LCS of c_i and c_j.

Definition 5.3 *Resnik similarity measure* (RES_{sim})

$$\text{rel}_{\text{RES}_{\text{sim}}}(c_1, c_2) = \max_{c_j \in \text{LCS}(c_1, c_2)} (\text{IC}(c_j))$$

In this definition RES is a similarity measure. The more similar two concepts, the more specific there LCS, and the higher their IC. It is easy to transform this measure to a distance measure, either by taking the probabilities instead of the log-probabilities, or by calculating the maximum IC, which is $-\log_2(\frac{1}{N})$ and subtracting the original IC. The *Information Vacuum* (IV) of a concept c_i is defined as

$$\text{IV}(c_i) = -\log_2(\frac{1}{N}) - \text{IC}(c_i) = \log_2(\text{freq}(c_i)) \ .$$

The RES measure can then be defined as a distance measure.

Definition 5.4 *Resnik distance measure* (RES_{dist})

$$\text{rel}_{\text{RES}_{\text{dist}}}(c_1, c_2) = \min_{c_j \in \text{LCS}(c_1, c_2)} (\text{IV}(c_j))$$

The higher the IC of the LCS of two concepts is, the lower is there IV, and the smaller is there distance.

The JCN (Jiang and Conrath, 1997) measure additionally uses the information content of the concepts that are compared. The distance between two concepts is defined as

Definition 5.5 *Jiang and Conrath distance measure* ($\mathrm{JCN}_{\mathrm{dist}}$)

$$\mathrm{rel}_{\mathrm{JCN}_{\mathrm{dist}}}(c_1, c_2) = \mathrm{IC}(c_1) + \mathrm{IC}(c_2) - 2(\mathrm{rel}_{\mathrm{RES}_{\mathrm{sim}}}(c_1, c_2)) \ .$$

With a similar reasoning as before the JCN measure can be converted into a similarity measure. The idea of this measure is that it is important to include the distance between the IC of the involved concepts. Since it is always sure that if $c_j \in \mathrm{LCS}(c_1, c_2)$, then $\mathrm{IC}(c_j) \leqslant \mathrm{IC}(c_1)$ and $\mathrm{IC}(c_j) \leqslant \mathrm{IC}(c_2)$ the definition does not lead to negative distances.

Word1	Word2	Similarity
paper	paper	1.000
paper	houses	0.476
paper	house	0.476
paper	writing	0.228
paper	activity	0.178
paper	equipment	0.143
paper	content	0.135
paper	division	0.129
paper	body	0.120
paper	people	0.111
paper	sports	0.109
paper	year	0.100
paper	airplane	0.100
paper	stay	0.099
paper	motor	0.096
paper	distance	0.092
paper	time	0.091
paper	bike	0.091
paper	mode	0.088
paper	comments	0.088

Table 5.2: 20 *highest JCN similarities for the noun 'paper' and other nouns in an N-best list history.*

Table 5.2 shows the 20 highest JCN similarities for the noun 'paper' and other nouns taken from N-best list histories in the RT05-S test data (Table 4.8). The word-sense disambiguation that is implicit in this comparison takes the senses of the words that maximizes the similarity (Definition 5.11).

The LIN (Lin, 1997, 1998) similarity measure uses the previous definitions in a different fashion.

Definition 5.6 *Lin similarity measure* (LIN_{sim})

$$\text{rel}_{\text{LIN}_{\text{sim}}}(c_1, c_2) = \frac{2(\text{rel}_{\text{RES}_{\text{sim}}}(c_1, c_2))}{\text{IC}(c_1) + \text{IC}(c_2)}$$

Suppose the LCS stays fixed, and the concepts one wants to compare get more specific, so there IC increases. In this case the similarity decreases, because the LCS gets more and more abstract in relation to the concepts. To increase the similarity the distance between LCS and concepts must decrease.

Path-based measures

The PATH measure uses the shortest path between two words in the WordNet graph. This measure is a distance measure. Two concepts are semantically more similar, the shorter the path between them. To convert this distance into a similarity measure, one just has to take the longest possible path, measured using the number of nodes, which is two times the depth D of the WordNet graph. The depth of the WordNet graph is the longest path from the root node to a leaf node. A leaf node is a node that has no children. The 'No' function returns the number of nodes in a path.

Definition 5.7 *Path similarity measure* (PATH_{sim})

$$\text{rel}_{\text{PATH}_{\text{sim}}}(c_1, c_2) = 2D - \text{No}(\text{shortestpath}(c_1, c_2))$$

The LCH (Leacock and Chodorow, 1998) measure also defines the length of a path using the number of nodes of the path. The length of the path between members of the same synonym set is therefore 1. This measure additionally scales by the maximum path length.

Definition 5.8 *Leacock and Chodorow similarity measure* (LCH_{sim})

$$\text{rel}_{\text{LCH}_{\text{sim}}}(c_1, c_2) = -\log_2\left(\frac{\text{No}(\text{shortestpath}(c_1, c_2))}{2D}\right)$$

The WUP (Wu and Palmer, 1994) measure uses the number of nodes $\mathrm{No}(\mathrm{path}(c_i, c_j)$ on the path between c_i and c_j, and the root node of the WordNet graph (root). It is defined as

Definition 5.9 *Wu and Palmer similarity measure* ($\mathrm{WUP}_{\mathrm{sim}}$)

$$\mathrm{rel}_{\mathrm{WUP}_{\mathrm{sim}}}(c_1, c_2) = \frac{2(\mathrm{No}(\mathrm{path}(\mathrm{root}, \mathrm{lcs})))}{\mathrm{No}(\mathrm{path}(c_1, \mathrm{lcs})) + \mathrm{No}(\mathrm{path}(c_2, \mathrm{lcs})) + 2(\mathrm{No}(\mathrm{path}(\mathrm{root}, \mathrm{lcs})))}$$

where $\mathrm{lcs} = \mathrm{LCS}(c_1, c_2)$. This measure takes into account the distance of the two sense nodes to their LCS, and the distance between the LCS and the root node. The similarity decreases when the distance between the sense node and their LCS increases, and it decreases also slowly with the increase of the distance between LCS and root node.

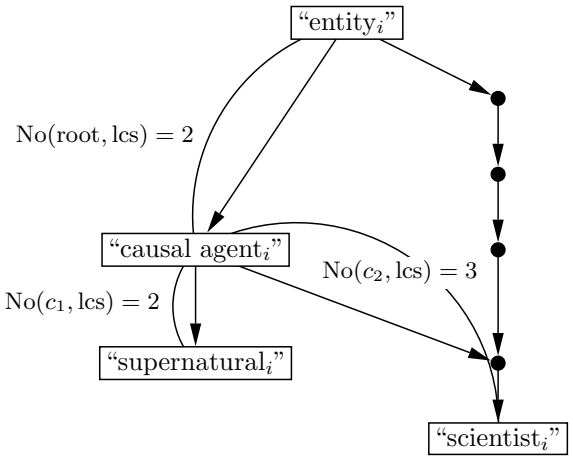


Figure 5.2: *Another subgraph of WordNet with LCS and root node.*

The graph in Figure 5.2 shows the necessary nodes and paths to measure the similarity of the concepts "supernatural$_i$" and "scientist$_i$" in this graph. The concept "causal agent$_i$" is the LCS of the two concepts, "entity$_i$" is the root node. The WUP-similarity between the two concepts is therefore $\frac{4}{9} = 0.444$. The minimum number of nodes in a path between two different nodes is 2, the minimum number of nodes in a path between a node and itself is 1. This assures that the definition is well defined. Otherwise it would not be defined between the root node and itself.

Measures for cross POS comparison

Relatedness measures that can compare words with different POS, like verbs and nouns, are especially useful. They can be used even with short contexts since they

can use all content words in the context. Some of them are also computationally very efficient.

The LESK measure was originally introduced by Lesk (1986) for word sense disambiguation. The basic idea is to define the semantic similarity between two concepts, by using the glosses/definitions of the concepts. Example 5.4 shows two glosses, one for the word 'train' and one for the word 'bus'.

(5.4)
- train – *public transport* provided by a line of railway cars coupled together and drawn by a locomotive.
- bus – a vehicle carrying many passengers; used for *public transport*.

The original LESK (Lesk, 1986) measure derives a semantic similarity score by simply counting the common word forms in the glosses. In this case the similarity score between these two senses of 'train' and 'bus' is 2. Function words (*a*, *by*, *of*, *and*, *for*) are ignored and can be filtered by providing a list of them. Remember that the basic semantic relatedness measures are defined on senses and not words or word forms. This is called the *matching coefficient* (Manning and Schütze, 1999, p. 299) and is defined as $\mid X \cap Y \mid$, where X and Y are the sets of all content words in the glosses.

Another possible way to make the comparison is the so-called *Jaccard coefficient*, which is used in Demetriou *et al.* (2000) for rescoring of simulated N-best lists. It is defined as $\frac{|X \cap Y|}{|X \cup Y|}$. For Example 5.4 the similarity value of the Jaccard coefficient is $\frac{2}{17} = 0.117$. This measure also takes into account the length of the glosses, and therefore does not boost the similarity of longer glosses. Manning and Schütze (1999, p. 299) gives three more definitions for semantic similarity by overlap of glosses/definitions.

The adapted LESK similarity measure that is used here, was introduced by Banerjee and Pedersen (2003). It extends the original measure in two ways. Taking again the example of 'train' and 'bus' one can see that they have a 2-gram in common, namely *public transport*. It is less likely for two definitions having a 2-gram in common than having a 1-gram in common.

Therefore one should give senses a higher similarity if they have a phrase overlap. This is taken into account in the extended LESK measure. This extended measure uses additionally not only the glosses of the compared senses, but also the glosses of related senses in the WordNet graph.

Table 5.3 shows the 20 highest LESK similarities for the adjective 'reliable' and other words taken from an N-best list history. Implicit word-sense disambiguation is achieved by taking those senses of the words that have the highest similarity (Definition 5.11).

Word1	Word2	POS2	Similarity
reliable	reliable	Adj.	1.000
reliable	better	Adj.	0.230
reliable	know	Verb	0.208
reliable	process	Noun	0.174
reliable	people	Noun	0.086
reliable	make	Verb	0.066
reliable	talk	Verb	0.062
reliable	were	Verb	0.053
reliable	are	Verb	0.053
reliable	going	Verb	0.050
reliable	was	Verb	0.048
reliable	am	Verb	0.048
reliable	is	Verb	0.048
reliable	good	Adj.	0.041
reliable	recant	Verb	0.039
reliable	okay	Adj.	0.034
reliable	red	Adj.	0.028
reliable	differs	Verb	0.025
reliable	first	Adj.	0.025
reliable	close	Adj.	0.023

Table 5.3: 20 *highest LESK similarities for the adjective 'reliable' and other words in an N-best list history.*

The HSO (Hirst and St-Onge, 1998) measure defines the strength of a relation between two words. It takes the length of an allowed path between two words (allowed-pathlength) and the number of changes of direction (dirchange) into account. A path can go through multiple relations like hypernymy, antonymy and so on, where each relation has a direction associated with it, which is either horizontal, down, or up. A path is allowed if it is of a pattern defined in Hirst and St-Onge (1998) and not longer than five links.

Extra-strong and strong relations get a fixed relatedness. An extra-strong relation holds between a word and itself. A strong relation holds between two words if they have a sense in common, or are horizontally related, or if one word is a phrase that includes the other and they are related. The similarity is defined in Definition 5.10 where C and k are constants. If there is no relation between two words, the relatedness

is 0. Definition 5.10 is taken from Budanitsky (1999).

Definition 5.10 *Hirst and St-Onge similarity* ($\mathrm{HSO_{sim}}$)

$$\mathrm{rel}_{\mathrm{HSO_{sim}}}(c_1, c_2) = \begin{cases} 2C & \text{if strong}(c_1, c_2), \\ 3C & \text{if extra\_strong}(c_1, c_2), \\ 0 & \text{if no relation}, \\ (C - \text{allowedpathlength} - k(\text{dirchange})) & \text{otherwise.} \end{cases}$$

5.2.2 Word context relatedness

Since the semantic relatedness of a word and a context shall be computed a word-context relatedness measure that uses the WordNet measures is defined. For the first evaluation a slightly modified version of the definition in Kozima and Ito (1995) is used. They used a semantic vector space, so they could directly define the distance between two words in context $\mathrm{dist}(w, w', C)$ by using the vectors in C to transform the original vector space and taking the Euclidean vector distance between w and w' in the transformed vector space. A context C is a multiset consisting of the previous δ words in the dialog, where δ is the context width. A multiset is a set, where an element can appear more than one time. A multiset can be represented as a standard set by attaching a counting index to each word that is added to the context. The cardinality of the multiset is the cardinality of such a representation. An ordered list could be used to also include the position of a word in the context.

For this work $\mathrm{rel}(w, w')$ is a relatedness between words, based on one of the WordNet-based relatedness measures between senses. For the following definitions of word-context, word-utterance-context measures and so on, a different definition results, depending on which basic relatedness measure between senses is applied.

Until now all relatedness measures were defined on concepts/senses/meanings. This is extended to words. $\mathrm{S}(w)$ are the senses of word w.[4]

Definition 5.11 *Word-word relatedness (similarity)*

$$\mathrm{rel}(w, w') = \max_{c_i \in \mathrm{S}(w)} \max_{c_j \in \mathrm{S}(w')} \mathrm{rel}(c_i, c_j)$$

In case of a distance measure the relatedness has to be minimized instead. The purpose of this definition is twofold. First it defines the semantic relatedness of two words. Second it defines word-sense disambiguation. It is the special case where just

[4] Later when it is necessary to determine the exact position of a word in a text, indices are used with words. Here it is not yet necessary since the measures are defined for words and word forms no matter where they are appearing. For the senses, indices are used, since they have a unique position in the WordNet hierarchy.

one word is used as a context for disambiguation (Pedersen *et al.*, 2005). Ideally the sense of a word in a text is also the sense that maximizes the semantic similarity of the word and the context, minimizes the semantic distance.

The relatedness of a word and a context ($\mathrm{rel_W}$) is defined as the average of the relatedness of the word and all words in the context.

Definition 5.12 *Word-context relatedness*

$$\mathrm{rel_W}(w, C) = \frac{1}{\mid C \mid} \sum_{w_i \in C} \mathrm{rel}(w, w_i)$$

The maximum similarity is used in Definition 5.11 when word-sense disambiguation is performed to find the senses of the words. In Definition 5.12 the average is used to take all words in the context into account, otherwise word repetitions always get the highest similarity.

5.2.3 Crossing part-of-speech (POS) boundaries

It is interesting to evaluate the measures that allow cross POS comparison separately. For these measures Definition 5.12 can be used directly. Since the width δ of the context does not change, the history that is taken into account is shorter. If just nouns are considered and the last δ nouns are found within the last k words in the history, k will be smaller if the last δ nouns and verbs are considered. k is then the length of the history that is spanned by the last δ content words. The number of verbs and nouns in an utterance is higher than the number of verbs or nouns alone. Consequently the history is shorter if cross POS comparison is used and δ is fixed.

This is an advantage or disadvantage depending on the application. If there is only a short context available, for example within a dialog system, the cross POS comparison is beneficial. For the case of conversational (telephone speech and meetings) speech, there is always a context that is long enough for all measures.

5.2.4 Word utterance (context) relatedness

The performance of the word-context relatedness (Definition 5.12) shows how well the measures work for algorithms that proceed in a left-to-right manner, since the context is restricted to words that have already been seen. For the rescoring of N-best lists it is not necessary to proceed in a left-to-right manner. The word-utterance-context relatedness can be used for the rescoring of N-best lists. This relatedness does not only use the context of the preceding words, but the whole utterance. In the case of rescoring of N-best lists the relatedness of a list element with the word-context or word-utterance-context is computed, and then the best combination of this score with the language model score is searched.

Suppose $U = \langle w_1, \ldots, w_n \rangle$ is an utterance. Let $\text{pre}(w_i, U)$ be the set $\bigcup_{j<i} w_j$ and $\text{post}(w_i, U)$ be the set $\bigcup_{j>i} w_j$. Then the word-utterance-context relatedness is defined as

Definition 5.13 *Word-utterance-context relatedness*

$$\text{rel}_{\text{U}_1}(w_i, U, C) = \text{rel}_{\text{W}}(w_i, \text{pre}(w_i, U) \cup \text{post}(w_i, U) \cup C) \ .$$

In this case there are two types of context. The first context comes from the respective dialog or monologue, and the second context comes from the actual utterance. To make an implementation of this definition efficient it is useful to cache already computed relatedness values. In Definition 5.13 the same relations are used multiple times. Another definition is obtained if the context C is eliminated ($C = \emptyset$) and just the utterance context U is taken into account.

Definition 5.14 *Word-utterance relatedness*

$$\text{rel}_{\text{U}_2}(w_i, U) = \text{rel}_{\text{W}}(w_i, \text{pre}(w_i, U) \cup \text{post}(w_i, U))$$

Both definitions can be modified for usage with rescoring in a left-to-right manner by restricting the contexts only to the preceding words.

Definition 5.15 *Word-utterance-context relatedness 1*

$$\text{rel}_{\text{U}_3}(w_i, U, C) = \text{rel}_{\text{W}}(w_i, \text{pre}(w_i, U) \cup C)$$

Definition 5.16 *Word-utterance relatedness 1*

$$\text{rel}_{\text{U}_4}(w_i, U) = \text{rel}_{\text{W}}(w_i, \text{pre}(w_i, U))$$

5.2.5 Defining utterance coherence

There are two basic types of connectedness of a text (Lyons, 1995, p. 263). Cohesion and coherence. Cohesion focuses on the form of a text considering the use of pronouns, particles, conjunctions and the like. Coherence deals with the connectedness of the content of a text. In Example 5.1 *they* and *them* in the second utterance refer to students and those people that torture them in middle and high school. In the context of this example these are probably teachers, since the second utterance also expresses that the students lost the interest in doing something in college, and teachers exist in middle and high school. If it turns out that the anaphora *they* and *them* have other referents, the text is less coherent or incoherent.

Concerning the senses of the words in Example 5.1, it can be derived from the assumption of coherence that the senses of the words 'school' and 'student' are related. The senses of the noun 'student' and some of the senses of 'school' in WordNet are

(5.5)
1. *student, pupil, educatee – a learner who is enrolled in an educational institution*
2. scholar, scholarly person, bookman, student – a learned person (especially in the humanities); someone who by long study has gained mastery in one or more disciplines

(5.6)
1. *school – an educational institution*
2. school, schoolhouse – a building where young people receive education
3. school – a body of creative artists or writers or thinkers linked by a similar style or by similar teachers
4. school, shoal – a large group of fish

If the word-word relatedness (Definition 5.11) is applied to the two words using the LESK measure, the highest relatedness is found between the first senses. These are the correct senses in this context, so the implicit word-sense disambiguation works for this example. The text/utterance is incoherent, if 'student' is used with the sense (1) in the first utterance of Example 5.1 and 'school' is used with sense (3) or (4).

This also shows a weakness of the word-sense disambiguation used here. If 'school' is used with the (4^{th}) sense, the disambiguation fails. The problem can be solved by taking a wider context for disambiguation.

Using Definitions 5.13-5.16 different concepts of utterance coherence can be defined, that cover parts of the linguistic concept of coherence. The coherences are not used in the first experiments where just the relatedness of each word to its context is computed. But for the rescoring they are used, when a score for each element of an N-best list is needed. U is again an utterance $U = \langle w_1, \ldots, w_n \rangle$. Since an utterance is represented as an n-tuple here, n can be taken as the cardinality of the utterance $\mid U \mid$.

Definition 5.17 *Inner-utterance-context coherence*

$$\text{coherence}_{\text{U}_1}(U, C) = \frac{1}{\mid U \mid} \sum_{w \in U} \text{rel}_{\text{U}_1}(w, U, C)$$

The first semantic utterance coherence measure (Definition 5.17) is based on all words in the utterance as well as in the context. It takes the mean of the relatedness of all words. It is based on the word-utterance-context relatedness (Definition 5.13).

Definition 5.18 *Inner-utterance coherence*

$$\text{coherence}_{\text{U}_2}(U) = \frac{1}{\mid U \mid} \sum_{w \in U} \text{rel}_{\text{U}_2}(w, U)$$

The second coherence measure (Definition 5.18) is a pure inner-utterance-coherence, which means that no history apart from the utterance is needed. Such a measure is very useful for rescoring, since the history is often not known or because there are speech recognition errors in the history. It is based on Definition 5.14.

Definition 5.19 *Utterance-context coherence*

$$\text{coherence}\mathrm{U}_3(U, C) = \frac{1}{\mid U \mid} \sum_{w \in U} \text{rel}_{\mathrm{U}_3}(w, U, C)$$

The third (Definition 5.19) and fourth (Definition 5.20) definition are based on Definition 5.15 and 5.16, that do not take future words into account.

Definition 5.20 *Utterance coherence*

$$\text{coherence}\mathrm{U}_4(U) = \frac{1}{\mid U \mid} \sum_{w \in U} \text{rel}_{\mathrm{U}_4}(w, U)$$

5.2.6 Relatedness to probability conversion

In language model adaptation for speech recognition, word-based n-gram models are often interpolated with other statistical language models to cover more syntactic, semantic and pragmatic information than is covered by the n-gram models alone (Bellegarda, 2004). To interpolate the Wordnet-based models with word-based n-gram language models, the relatedness functions must be converted to conditional probabilities. Here this probability is defined for a given vocabulary V as

Definition 5.21 *Relatedness to probability conversion*

$$p_{\text{wordnet}}(w_1 \mid w_2) = \frac{\text{rel}(w_1, w_2)}{\sum_{i=1}^{|V|} \text{rel}(w_i, w_2)} \ .$$

The conditional probability of a word given a context (e.g. multiple words) can be easily derived from the above definition. Since the computation of the relatedness for all words in the vocabulary is computationally expensive, it is most efficient to approximate $\sum_{i=1}^{|V|} \text{rel}(w_i, w_j)$. This can be done with a Monte Carlo method by randomly selecting a portion of the vocabulary, computing the mean relatedness and deriving the total relatedness from the mean. For each word w_j in the vocabulary one sum has to be estimated. If V is the vocabulary and $\mid V \mid = k$ then k sums need to be estimated.

Our experiments (Pucher and Huang, 2005; Pucher *et al.*, 2006a,b) and other work on LSA-based models (Deng and Khudanpur, 2003; Coccaro and Jurafsky, 1998), which basically also rest on a concept of semantic relatedness (the cosine similarity),

show that linear interpolation of semantic and n-gram models is not useful. Therefore one can try for example a log-linear interpolation. Suppose that $p_{n\text{-gram}}$ is the n-gram model, and $\propto$ denotes normalization over the whole vocabulary. The log-linear interpolation can be defined as

Definition 5.22 *Log-linear interpolation of WordNet and n-gram models*

$$p(w \mid C) \propto p_{\text{wordnet}}(w \mid C)^{\lambda}\ p_{n\text{-}gram}(w \mid C)^{1-\lambda}\ .$$

The normalization term is given by the sum $\sum_{i=1}^{|V|} p_{\text{wordnet}}(w_i \mid C)^{\lambda}\ p_{n\text{-gram}}(w_i \mid C)^{1-\lambda}$. Even if restricting the context to one word, which one can do since the WordNet probability can be broken down to conditional probabilities of a word given another word, it is necessary to estimate the sum for each word given any other word. Starting with k sums, it is now necessary to estimate k^2 sums. At this stage it is easier to estimate the k^2 word-word similarities and start from there. This is however computationally very expensive given a vocabulary of around $40,000$ words.

Another possibility is to use the maximum entropy framework to estimate conditional probabilities. Within the conditional maximum entropy framework it is possible to create a language model using a mixture of features (Rosenfeld, 1994). A union of n-gram and Wordnet-based semantic features is a good starting point. A possible WordNet feature, which is an adaptation of an LSA feature defined in Deng and Khudanpur (2003), is

Definition 5.23 *WordNet relatedness feature*

$$f_{\text{wordnet}} = \begin{cases} 1 \text{ if } \text{rel}_{\text{W}}(w, C) > \theta \text{ and words are comparable,} \\ 0 \text{ otherwise.} \end{cases}$$

The condition that the words be comparable reflects that these measures can only be applied to content words, and that certain measures are more useful than others. This condition can be checked using POS tagging. Nouns for example should be compared with nouns; verbs and adjectives with nouns, verbs and adjectives. The expectation of this feature ($\mathbb{E}[f_{\text{wordnet}}]$) can be derived from a large training corpus. If this feature is taken with a realistic context width of 5–10 words, it is likely that two instances of the same word will not appear in the same context.

Since the contexts C for different words are almost surely different, one has to estimate a separate parameter for each word in the training corpus. For each word w in the training corpus, $\text{rel}_{\text{W}}(w, C)$ needs to be evaluated, which is again computationally very expensive. One solution to this problem is to restrict the size of the context to the previous, or the two previous content words. This results in a trigger-based approach where word trigger pairs or triples are defined. With LSA-based features it

is possible to reduce the number of contexts by clustering the document space (Deng and Khudanpur, 2003).

So one viable approach is to skip the conversion into a probability distribution and apply the relatedness measures directly to N-best lists, as in Demetriou *et al.* (2000).

5.3 Experiments

5.3.1 Performance measuring

To estimate the word prediction performance of the WordNet measures, the method described in Kozima and Ito (1995) is used. This performance metric is used, because it allows for a comparison of differently scaled semantic relatedness measures. To measure the word prediction performance for a word w_p in a context C, the vocabulary $V = \{w_1, \ldots, w_n\}$ of the whole dialog is ordered according to context relatedness, such that

Definition 5.24 *Word-context relatedness ordering*

$$\mathrm{rel_W}(w_{i_1}, C) > \mathrm{rel_W}(w_{i_2}, C) > \cdots > \mathrm{rel_W}(w_{i_n}, C)$$

is an ordering of all n words of the vocabulary V, given the context C. For distance measures the ordering has to be reversed. Suppose i_p is the position of word w_p in this ordering. The performance for w_p is

Definition 5.25 *Performance measure*

$$\mathrm{perf}(w_p) = \frac{\mid V \mid /2 - i_p}{\mid V \mid /2} \ .$$

If the word w_p occurs in the first half of the ordered vocabulary, the performance score is positive. If it occurs in the second half it is negative. The performance scores are between -1 and 1. If i_p is randomly selected the mean score is 0. A positive mean score shows that a relatedness measure performs better than random on the task of word prediction. Using this metric, performance scores for the eight different WordNet-based relatedness measures that were previously defined are computed.

5.3.2 Evaluation results for word-context relatedness

For the first evaluation cross part-of-speech comparison is not used, the Brown corpus is taken for the information content files that are needed for the LIN, RES, and JCN measures, and the whole dialog (both speakers in each dialog of the CallHome corpus) is taken as the context.

Rel.	POS	Performance	Mean perf.
JCN	N	0.387	0.385
JCN	V	0.383	
WUP	N	0.299	0.313
WUP	V	0.328	
PATH	N	0.333	0.307
PATH	V	0.281	
LESK	N	0.299	0.288
LESK	V	0.277	
RES	N	0.290	0.288
RES	V	0.286	
HSO	N	0.254	0.227
HSO	V	0.201	
LCH	N	0.250	0.225
LCH	V	0.200	
LIN	N	0.220	0.194
LIN	V	0.169	

Table 5.4: *Word-context relatedness performance.*

The context is restricted to a context-width of $\delta = 5$ in all performance evaluations. So the last 5 content words of the whole dialog are taken as the context in these first experiments. Since the context is that short a decay parameter is not used. Such a parameter gives words in the recent history a higher weight. Five dialogs from the CallHome English corpus are used, and the average performance value for verbs and nouns is computed. Tests with $\delta = 10$ and $\delta = 15$ showed that the average scores are slightly higher, but the ranking of the measures did not change. Since the reason for these performance evaluations is to give a ranking of the measures and due to the computationally expensive reordering of the whole vocabulary, the lowest context-width is taken.

The evaluation tables contain the name of the relatedness measure, the part-of-speech (N for noun and V for verb), and the mean performance values for each part of speech. Cross POS comparison is not used so the lines in Table 5.4 give the performance for nouns when predicted from the noun context and verbs when predicted from the verb context. The last column contains the mean performance value for nouns and verbs together and determines the ranking of the measures. As one can see from Table 5.4 the measure JCN has the best overall performance, followed by the measures WUP and PATH which are based on path lengths (See Section 5.3.6 for

significance results).

It is surprising that the WUP and PATH measures, which are only based on path lengths have such good performance scores compared to the RES measure that also includes corpus information. The good performance of the JCN measure was already reported by Budanitsky and Hirst (2001) for the task of malapropism correction.

Figure 5.3 shows the mean performance of some measures for the five dialogs, for all dialogs, and for noun and verb measures together. It can be seen that the performance varies from dialog to dialog but the ranking of the measures is quite stable. Since the performance score is always bigger than zero, we can conclude that the measures perform better than random.

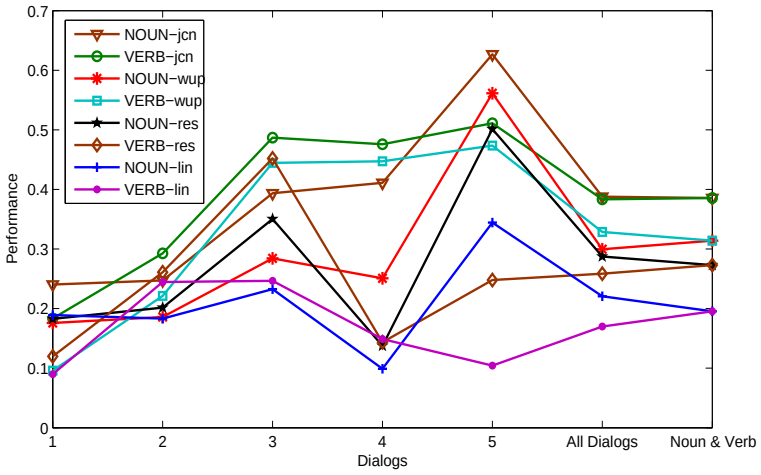


Figure 5.3: *Word-context relatedness performance.*

In Figure 5.3 one can also see that there is a performance pattern. All measures have lower performance scores on the first two dialogs and a higher score on the fifth dialog (Except for the VERB-lin and VERB-res measure). The NOUN-jcn and VERB-jcn measures are the top performers under the noun and verb measures for all the dialogs.

5.3.3 Evaluation results for crossing POS boundaries

Table 5.5 shows the results for the two measures that allow cross part-of-speech comparison, namely HSO and LESK, for which the context contains nouns and verbs.

The LESK measure performs very well for verbs and performs worse than random for nouns, which gives an overall performance score of 0.212. The prediction of verbs from a context containing nouns and verbs performs better than from a context containing only verbs, which can be seen by comparison with the performances of verb measures in

Rel.	POS	Performance	Mean perf.
LESK	N	-0.002	0.212
LESK	V	0.427	
HSO	N	0.068	0.186
HSO	V	0.305	

Table 5.5: *Word-context relatedness performance across POS.*

Table 5.4. Especially the LESK measure for verbs using only a verb context performs worse (0.277 versus 0.427).

The prediction of nouns from a mixed context performs worse than from a context containing only nouns, which is shown in the noun column of LESK in Table 5.5. This is again true for all noun measures that use only noun contexts (Table 5.4).

The HSO measure also performs worse for nouns when using a mixed context (see Table 5.5) than only a noun context, and better for verbs when using a mixed context. For verb prediction using a mixed measure also HSO is outperformed by the LESK measure.

So it is shown that the relatedness of nouns to a noun context is highest and the relatedness of verbs to nouns and verbs is highest using the LESK measure. This measure is based on extended WordNet glosses. It should be possible to explain the facts concerning performance using facts about WordNet glosses or lexical semantics in general. One possible explanation is that nouns exhibit a strong internal ordering and hierarchy, which is exemplified by the hypernym-hierarchy. Verbs are lacking such an ordering.

5.3.4 Evaluation results for word-monologue-context relatedness

This section evaluates the performance of the measures for different dialog contexts. The word-context relatedness is used, and the context is restricted to the monologues (The performance of word-context relatedness for the whole dialog is already shown in Table 5.4).

As Table 5.6 shows, the performance decreases in general when using just the monologue context, relative to Table 5.4, which uses the whole dialog as a context. Only the JCN measure still performs quite well when using just the monologue.

The good performance of JCN is somewhat surprising, when assuming that the semantic coherence across the whole dialog is higher than just within the monologue. Partly it can be explained by bigger monologue chunks in the dialogs as can be seen in Example 5.1 and 5.2.

Rel.	POS	Performance	Mean perf.
JCN	N	0.334	0.315
JCN	V	0.297	
PATH	N	0.256	0.222
PATH	V	0.188	
LCH	N	0.249	0.217
LCH	V	0.186	
LESK	N	0.237	0.210
LESK	V	0.183	
HSO	N	0.230	0.200
HSO	V	0.171	
RES	N	0.214	0.183
RES	V	0.153	
LIN	N	0.192	0.167
LIN	V	0.143	
WUP	N	0.184	0.164
WUP	V	0.144	

Table 5.6: *Word-monologue-context relatedness performance.*

5.3.5 Evaluation results for word-utterance-context relatedness

In this section Definition 5.13 is used to measure the performance of word prediction using a context and an utterance.

When using this measure the context width is $\delta = 5$ plus the number of verbs/nouns in the utterance the word belongs to. In the experiments in Section 5.3.2-5.3.4 only the previous $\delta = 5$ content words were used. These content words could come from the actual utterance or previous utterances. So one expects the performance to be better with this measure since it uses a wider context. Concerning the overall score of the best measure this is true. The ordering of the measures slightly changes. The PATH and RES measure perform better for nouns and verbs, and the LESK measure performs better for verbs with this definition. The JCN measure still performs well under this condition (See Section 5.3.6 for significance results).

Rel.	POS	Performance	Mean perf.
PATH	N	0.392	0.398
PATH	V	0.405	
JCN	N	0.367	0.367
JCN	V	0.368	
RES	N	0.344	0.356
RES	V	0.369	
LESK	N	0.247	0.295
LESK	V	0.343	
LCH	N	0.265	0.236
LCH	V	0.207	
HSO	N	0.248	0.221
HSO	V	0.195	
LIN	N	0.220	0.194
LIN	V	0.169	
WUP	N	0.206	0.184
WUP	V	0.163	

Table 5.7: *Word-utterance-context relatedness performance.*

5.3.6 Performance comparison

Noun measures

T-tests for paired samples indicate that the performance values of the PATH measure using the word-utterance-context and the whole dialog (*path_d_s*) are significantly higher ($p < .05, \#\text{nouns} = 722$) than all other noun-related measures, except the JCN measure using the word-context and the whole dialog (*jcn_d_w*). Revealing the second-highest mean performance, *jcn_d_w* performs significantly better than most other noun-related measures (paired samples t-tests; $p < .05, \#\text{nouns} = 722$), except *path_d_s*, *res_d_s*, JCN using the word-context of the monologue (*jcn_m_w*) and JCN using the word-utterance-context (*jcn_d_s*). There is no significant difference between *path_d_s*, *jcn_d_s* and *res_d_s*.

For this reason we can conclude that the JCN measure should be used for the rescoring task, since it is the most robust measure. For the word-context case it is significantly better than all other word-context based measures. The same is true for the word-monologue-context case. For the word-utterance-context case there is no significant difference between the three best performing measures, so the JCN measure can also be used with this condition.

Verb measures

According to t-tests for paired samples, LESK using a mixed word-context of the dialog (*lesk_cross_d_w*) performs significantly better than all other verb-related measures ($p = .05, \#\text{verbs} = 597$), except *path_d_s*, which has the second-highest mean performance value.

Although the LESK measure does not outperform the PATH measure under certain conditions, there are still reasons to prefer the LESK measure. The first is computationally less expensive than the second. Furthermore it can be used, at least the simple version of it, with dictionaries other than WordNet. For these reasons the *lesk* measure is used for the rescoring task.

5.3.7 Word-error-rate (WER) experiments

For the rescoring experiments test data from the NIST Rich Transcription 2005 Spring (RT-05S) meeting evaluation (Table 4.8) is used. N-best lists are generated using the combined n-gram language model (Table 4.21). The first-best element of the previous N-best list is added to the context, like in the rescoring with LSA-based models (Section 4.5.6).

Before applying the WordNet-based measures, the N-best lists are POS tagged with a decision tree tagger (Schmid, 1994). The WordNet measures are then applied to verbs, nouns and adjectives. Then the similarity values are used as scores, which have to be combined with the language model scores of the N-best list elements.

For reasons given in the preceding section two different measures are used for verbs and nouns. The JCN measure is used for computing a noun score based on the noun context, and the LESK measure is used for computing a verb/adjective score based on the noun/verb/adjective context. In these experiments adjectives are also included. In the end there is a *lesk_score* and a *jcn_score* for each N-best list element.

Since a type of log-linear interpolation performed best in the LSA rescoring experiments, a log-linear interpolation method is used for the rescoring based on WordNet, too. As mentioned in Section 5.2.6 it is computationally expensive to transform a relatedness measure to a probabilistic measure. Based on all the language model scores of an N-best list a probability is estimated, which is then interpolated with the n-gram model probability as described in Definition 5.22. If only the elements in an N-best list are considered, log-linear interpolation can be used since it is not necessary to normalize over all sentences. Thereby one can reduce the computational complexity of the normalization. Then there is only one parameter λ to optimize, which is done with a brute force approach. For this optimization a small part of the test data is taken and the WER is computed for different values between 0 and 1.

In Section 5.3.4 it has been shown that using only the monologue context results in a performance decrease. For this reason the whole dialog context is used for rescoring.

Furthermore different utterance and dialog context measures are applied. Finally the following models are used in the WER experiments.

As a baseline the n-gram mixture model trained on all the available training data (Table 4.7) is used. It is log-linearly interpolated with the WordNet probabilities. Additionally to this sophisticated interpolation, solely the WordNet scores are used without the n-gram scores.

WER experiments for inner-utterance coherence

In this first group of experiments Definitions 5.17 and 5.18 are applied to the rescoring task. Similarity scores for each element in an N-best list are derived according to the definitions. The first-best element of the last list is always added to the context. The context size is constrained to the last 20 elements. Definition 5.17 includes context apart from the utterance context, Definition 5.18 only uses the utterance context.

No improvement over the n-gram baseline is achieved for these two measures. Neither with the log-linearly interpolated models nor with the WordNet scores alone. The differences between the methods in terms of WER are not significant.

WER experiments for utterance coherence

In the second group of experiments Definitions 5.19 and 5.20 are applied to the rescoring task. There is again one measure that uses dialog context (5.19) and one that only uses utterance context (5.20).

Also for these experiments no improvement over the n-gram baseline is achieved. Neither with the log-linearly interpolated models nor with the WordNet scores alone. The differences between the methods in terms of WER are also not significant. There are also no significant differences in performance between the second group and the first group of experiments.

5.3.8 Analysis of WER experiments

From the WER experiments one can derive that word prediction performance as it is defined and measured in this chapter is no good indicator for WER performance. One reason for the poor performance of the WordNet models on this task could be the low WordNet coverage of the N-best lists.

For the different meeting sites (AMI, CMU, ICSI, NIST, VT) the percentage of content words (nouns, verbs, and adjectives) in the N-best lists covered by WordNet lies between 89% and 92%. To measure this the content words of the POS-tagged N-best lists are extracted and looked up in WordNet. For content words these numbers are also influenced by the quality of the POS tagging. For different meetings we get the

same percentage range. The WordNet coverage for different meetings is between 89% and 92%.

For different speakers the coverage of content words is between 87% and 100% with the exception of four speakers that are shown in Table 5.8. The third column contains the number of content word tokens in the speaker N-best list that are in WordNet, the fourth column shows the total number of content words in the N-best lists of the speaker. So there are two speakers that have one N-best list containing two content words, where one content word is in WordNet.

Speaker ID	Coverage in %	CWord Tokens in WN	Total CWord Tokens
NIST-102	50	1	2
NIST-103	50	1	2
CMU-fYRQGJN	75	57,798	76,145
ICSI-mn007	78	16,091	20,537

Table 5.8: *Content words in N-best lists covered by WordNet.*

The WordNet coverage of content words can therefore be no reason for the low performance of the WordNet models.

If all words are counted the picture is different. For different meeting sites (AMI, CMU, ICSI, NIST, VT) the percentage of words in the N-best lists covered by WordNet lies between 47% and 48%. At the level of meetings (2 meetings per site) the percentage of covered words lies between 46 and 49%.

Especially the NIST and ICSI meetings contain speakers that are not well covered by WordNet. For NIST these are speakers that have very short contributions to the meeting between 17 and 19 words.

The WordNet coverage for ICSI speakers is shown in Table 5.9. There are a few speakers that have a very low coverage (25 to 27%). But these few speakers cannot be the reason for the overall low performance of the WordNet models. What can be a problem however, is the low overall coverage of N-best lists.

Another reason for the poor performance of the models could be that the task of rescoring simulated N-best lists, as presented in Demetriou *et al.* (2000), is significantly easier than the rescoring of 'real' N-best lists. In the above chapter it was also shown that WordNet models can outperform simple random models on the task of word prediction. In the above WER experiments furthermore a 4-gram baseline model was used, which was trained on nearly 1 billion words. In Demetriou *et al.* (2000) a simpler baseline has been used. 650 sentences were used there to generate sentence hypotheses with different WERs using phoneme confusion data and a pronunciation lexicon. Experiments with simpler baseline models ignore that these simpler

Speaker ID	Coverage in %	Word Tokens in WN	Total Word Tokens
fn002	25	26	101
me006	26	79	300
mn014	27	7,761	28,187
mn007	33	23,234	68,401
me001	41	138,023	328,738
mn017	43	288,100	668,537
me022	45	284,376	601,369
me026	47	104,850	218,891
me013	48	1,116,548	2,319,871
fe060	50	364,659	728,637
fe016	50	192,778	379,919
me018	50	257,416	507,048
me011	51	225,978	440,917

Table 5.9: *ICSI speaker words in N-best lists covered by WordNet.*

models are not used in today's recognition systems.

5.4 Summary and discussion

In this chapter different WordNet-based relatedness measures are introduced. We show how to define more and more complex relatedness measures on top of the basic relatedness measures between word senses. These complex relatedness measures are defined on different contexts.

A ranking of the usefulness of semantic relatedness measures for word prediction in conversational speech is given. It is shown that there are significant differences in the performance of these measures and that all measures perform better than random on this task.

The JCN measure performs best for nouns using the word-context of the whole dialog or the monologue, and the PATH measure performs best for nouns using the word-utterance-context of the dialog. The same result for the JCN measure was obtained by Budanitsky and Hirst (2001) for a different task. The LESK measure performs best for verbs using a mixed word-context. It can be concluded that different measures should be used for the prediction of nouns and verbs, and for different contexts. Since the JCN measure shows the best overall performance for nouns using a noun context it is used in the WER experiments. For verbs and adjectives the LESK measure is used with a mixed context.

The insight on when to use a mixed content word context and when not, is also in accordance with counting arguments about the classes of content words. Nouns can be thought of as an independent hierarchy. Verbs and adjectives are used together with nouns. Verbs and adjectives have many different nouns they can be used with, while a certain noun has a smaller number of verbs and adjectives it can be used with. Therefore it is easier to predict a noun from a noun context and a verb or adjective from a noun/verb/adjective context.

These results were the basis for an investigation of the use of the best performing measures for the task of speech recognition hypotheses rescoring for multi-party meetings. The LESK and JCN measures were used for the rescoring of N-best lists. It was shown that speech recognition of multi-party meetings cannot be improved compared to a 4-gram baseline model, when using WordNet models. The analysis concerning WordNet coverage showed that the content words are well covered in WordNet. The coverage of all words is however smaller.

Other researchers (Demetriou *et al.*, 2000) who reported improvements of WER using these models used simpler baseline models and synthetic N-best lists. We think that it is not useful to base a comparison on simpler baselines when much larger models are already available and actually used.

We think that these prediction models can still be useful for other tasks where only small amounts of training data are available. Another possibility of improvement is to use other interpolation techniques like the maximum entropy framework. WordNet-based models could also be improved by using a trigger-based approach. This could be done by not using the whole WordNet and its similarities, but defining word-trigger pairs that are used for rescoring.

6 Empiricist and rationalist modeling paradigms

> Those who have treated of the sciences have been either empiricists or dogmatists. Empiricists, like ants, simply accumulate and use; Rationalists, like spiders, spin webs from themselves; the way of the bee is in between: it takes material from the flowers of the garden and the field; but it has the ability to convert and digest them (Bacon, 2002, p. 79).

Latent Semantic Analysis (LSA) and WordNet-based models can be subsumed under the concept of 'semantic similarity models'. The underlying modeling paradigms are however different. Approaches in speech and language processing and other fields of *Machine Learning* (ML) can be divided into rationalist and empiricist approaches, in analogy to rationalist and empiricist philosophical theories of human knowledge and learning.

6.1 Empiricism and rationalism

Knowledge can be defined as justified, true belief, although it is known that examples can be constructed where these three properties are not sufficient for knowledge (Gettier, 1963). Rationalists and empiricists have different views on how we can gain knowledge. According to Markie (2006) a rationalist must adopt at least one of the following claims. The Intuition/Deduction Thesis, the Innate Knowledge Thesis or the Innate Concept thesis.

> The Intuition/Deduction Thesis: Some propositions in a particular subject area, S, are knowable by us by intuition alone; still others are knowable by being deduced from intuited propositions.

The above thesis states that we have access to *a priori* knowledge by intuition. From that knowledge we can use valid reasoning to derive other propositions. A subject area for which the Intuition/Deduction thesis has been adopted is mathematics. A proposition that Descartes believed to be known by intuition like other mathematical propositions is the geometric proposition that the three angles of a triangle are equal to two right angles (Descartes, 1996, p. 45).

> The Innate Knowledge Thesis: We have knowledge of some truths in a particular subject area, S, as part of our rational nature.

The Innate Knowledge Thesis also states that we have some *a priori* knowledge. The difference to the former thesis is that we do not gain this knowledge by intuition, but have it as part of our nature.

> The Innate Concept Thesis: We have some of the concepts we employ in a particular subject area, S, as part of our rational nature.

The Innate Concept Thesis is equivalent to the former thesis except that it states that some concepts we have are *a priori*.

The empiricist on the other hand adopts the following claim:

> The Empiricism Thesis: We have no source of knowledge in S or for the concepts we use in S other than sense experience.

Since the claims are all relativized to a certain domain S, there is only a conflict if rationalists and empiricists want to provide a foundation for the same domain. It is consistent to be a rationalist in mathematics, and an empiricist in the natural sciences. The most debated type of knowledge is our knowledge about the external world.

Manning and Schütze (1999, p. 4) sees a dominance of empiricist approaches between 1920 and 1960 followed by a rationalist period influenced by the work of Noam Chomsky and others, followed again by an empiricist period that gained momentum in the last decades. The recent rise of empiricist paradigms is related to the availability of increasing amounts of data and more and more computing power to train models on these data.

The rationalist and empiricist approach to language can be characterized by the problem of the *poverty of the stimulus* (Chomsky, 1986, p. 7) as seen by the rationalist or the contrary *richness of the stimulus* as seen by the empiricist. The sensory data that are the stimuli do not provide enough information to account for all language capabilities, says the rationalist. Therefore the rationalist assumes that a part of human knowledge about language is fixed in advance and cannot be derived from the senses.

The empiricist is maybe surprised how the sensory data can account for our highly complex picture of the world, but nevertheless thinks that there must be a way to derive this picture from the sensory data without assuming too much fixed knowledge. According to the empiricist there are only general cognitive abilities at the beginning and most of our knowledge is derived by the senses. Concerning the cognitive capabilities that are present in the brain empiricism is only gradually differing from rationalism. Empiricism also assumes some cognitive capabilities as present in the brain, like association, pattern recognition, and generalization (Manning and Schütze, 1999, p. 5). Chomsky (1969) observes that Quine (1960, p.83-84) assumes a prelinguistic

quality space with a built-in distance measure in his empirical theory of learning. Depending on the nature of this quality space one can come up with different rationalist theories of innate ideas.

In this thesis two different modeling paradigms are applied to the same task. The advantages and disadvantages of each approach are discussed. But which analogies justify us to label one method 'empiricist' (LSA) and the other method 'rationalist' (WordNet)?

6.2 Rationalist WordNet-based models

WordNet (Fellbaum, 1998, p. 8) is a graph of lexical semantic relations between word senses. The data was collected by a group of experts.

Rationalism must adopt at least one of the three theses mentioned in Section 6.1. With an ontology[1] like WordNet it is clear that not all concepts in WordNet can be thought of as innate (Innate Concept Thesis) because that would make learning impossible. Neither is it possible to think of all propositional knowledge encoded in WordNet as innate (Innate Knowledge Thesis) or knowable by intuition and deduction (Intuition/Deduction Thesis).

It does not make sense to have an innate concept of "vacuum cleaner" or propositional innate knowledge about vacuum cleaners, because if they would not have been invented, we would still have the innate concept. But concerning the three theses from the previous section the models based on WordNet can be called 'rationalist' if there is a part of WordNet, be it concepts or propositions, that can be identified as innate, including all the knowledge that can be derived from it. Another analogy to rationalism is that the WordNet graph has been created by human experts, and has not been directly derived from data.

The analogy to the rationalist understanding of human knowledge is that not all information comes from sensory data, but there is some structure hard-wired into the perceivers to put structure on the sensory data.

The main disadvantage of the WordNet-based models and rationalism is that the structure must be given somehow, either through the work of human experts in the case of WordNet-based models for language modeling, or through a super-human expert in the case of rationalist models of human knowledge.

In Chapter 5 these rationalist structures are reduced to a binary relation of 'similarity'. Thereby a lot of information, especially about the inter-connectedness of concepts

[1] In the philosophical sense an ontology describes the things that really exist, and their relations (Quine, 1953a). In this sense WordNet is not an ontology, since it is derived from word usage, and it is not necessary that all words that are used have irreducible referents. WordNet for example contains the concept of a supernatural being which is a subclass of "causal agent". WordNet is neutral concerning ontological decisions, it contains physical objects as well as abstract objects.

is lost. This reduction is necessary for the WordNet-based models to be integrated into the speech recognition framework and be comparable with the empiricist LSA models. If the 'empiricist' LSA model outperforms the 'rationalist' WordNet model it is not a definitive judgment against the latter, since the experimental setting does not make use of all the information that is encoded in the WordNet graph.

6.3 Empiricist latent semantic analysis (LSA) based models

In LSA-based modeling a word-document co-occurrence matrix is used for the estimation of semantic relatedness between words and documents. The documents that contain word forms can be considered as sensory data. In this case the data is not without any structure, since it is already structured into words and documents. But it has less structure than WordNet. Therefore it is justified to call the LSA modeling approach 'empiricist'. The documents also contain a linear structure where one word is followed by another word. This linear structure is however not used by LSA models. Documents are treated as bag-of-words.

The epistemological analogy concerning human understanding is that the only available data for our knowledge of the world is almost unstructured sensory data, and all knowledge is somehow derived from that sensory data.

The less structure there is in the data the more 'empiricist' the modeling approach. For ML one can however take as much structure as is available. If, for example, there are also topic boundaries additionally to the document boundaries, these can also be taken into account. This leads to a continuum from empiricist to rationalist modeling approaches.

The advantage of these types of models is that almost no structure needs to be given for the emergence of semantics. The interesting question is how something can be learned from the data. In Section 4.3 it is shown that LSA models can learn synonymy to a certain extent, and references to investigations are given that show how similar methods can learn other semantic concepts.

7 Conclusion

7.1 Summary

The main results of this thesis are:

1. In Subsection 4.2.5 a new method for combining multiple LSA models has been introduced - The model combination is reasonably fast to combine large models. For the interpolation of all LSA models no improvement in terms of perplexity or WER has been achieved.

2. In Section 4.4 an extensive analysis of LSA-based models has been conducted, including the optimization of parameters for combining multiple LSA models - Different optimization parameters have been analyzed, which have been introduced after LSA models were used in language modeling for the first time. The model analysis also has shown that LSA models cover semantic concepts like synonymy to a certain extent.

3. In Section 4.5 LSA-based language models trained on the basis of different background domain data have been used in ASR for multi-party meetings - Significant improvements in terms of perplexity and WER over the baseline n-gram models have been achieved.

4. Subsection 5.2.4-5.2.5 defines new word-utterance context measures and new utterance coherence measures - Here it has been shown how different context measures can be defined for rescoring of N-best lists. These context measures can be used with semantic relatedness measures, like the ones based on WordNet applied in this work.

5. Subsection 5.3.1-5.3.6 contains an evaluation of WordNet-based relatedness measures for word prediction in conversational speech - The evaluation has shown that different measures and contexts should be used for nouns and verbs. Furthermore it has been shown that WordNet-based models can outperform simple baseline models for this task.

6. Subsection 5.3.4 contains an evaluation of WordNet-based relatedness measures using the monologue context for word prediction in conversational speech - Here it has been shown that using the monologue context performs worse than using the whole dialog context.

7. Subsection 5.3.7-5.3.7 contains results on the application of different WordNet-based context measures in ASR for multi-party meetings - It has been shown how these measures can be applied for the recognition task. No improvements in terms of WER have been achieved.

The overall results concerning LSA and WordNet models are similar. If the models are applied to simple tasks (perplexity or word prediction) and if they are compared to simple baselines, both modeling approaches are able to outperform the baseline models (n-gram model and random model). But concerning better baseline models, like the large n-gram model trained on all available data ($\approx$ 1 billion words), and more complex tasks, like ASR for multi-party meetings, the models do not achieve an improvement.

From that it can be derived that n-gram models somehow cover the semantic information contained in the semantic models, if enough training data is available. It makes a significant difference what the training data size of a language model is. It can happen that for a small training corpus a modeling method outperforms n-grams, but for a larger corpus this gain is lost.

In this work 4-gram models were used. If more data is available higher n-gram models can be used. The most significant improvement is however seen between 1 and 2-gram models, and between 2 and 3-gram models. Between 3 and 4-gram models the improvement is already smaller. So we think that one has to use at least 3-gram models to observe the above mentioned effect.

Concerning the size of the training data set one criticism should be mentioned here. If our language modeling approaches are to be related with human speaker abilities of word prediction, this relation is lost with the brute-force approach of using more and more training data. Given the 837 million word tokens that were used for training the 4-gram models, a 20-year old human speaker would have to listen or read 1.33 words each second, if this database is taken as an empirical basis for language learning. This is clearly impossible. So human language learning is very efficient concerning the amount of seen data. In terms of learning time however, human language learning is less efficient. An n-gram model can be trained within a few hours, while humans need a much longer training time.

So the dilemma is that there is a simple modeling[1] approach that has little relation to human language learning, and there are more sophisticated modeling approaches that cannot outperform the simple approach, if enough data is available. Either one uses the sophisticated approach, and can include linguistic insights, or one uses the simpler approach and has a robust high-performance model.

What we are still lacking in my opinion is a language modeling method that can outperform word-based n-gram models on very large training data sets.

[1] Sophisticated smoothing methods are applied in word-based n-gram language modeling. In this respect the approach is not simple.

7.2 Future work

A number of issues are presented, which deserve further analysis. During the work on this thesis and the review of the literature I have identified other lines of work that may be worth investigating.

- One interesting domain for future work could be 'LSA model analysis'. The cosine similarity could be used to measure the distance and correlation of models where a close correlation means that the semantic spaces of the models are close. From that follows another direction for future work on model training by selecting appropriate web data that fits well the domain. This could be achieved by selecting web data based on n-gram counts, training an LSA model with this web data, and comparing this model with the domain model.

- Concerning the WordNet models it would be interesting if the performance results for different contexts and measures are valid in other domains of research like information retrieval.

- It would also be interesting to know how n-gram language models cover linguistic information. It is known that n-gram models cover linguistic relations at all levels of linguistics. Two questions arise. Exactly which linguistic relations are covered by n-grams? What is the relation between these linguistic relations, the training data size, and the size of the n-gram context n?

- Since no improvements over the large 4-gram baseline have been achieved with neither LSA nor WordNet models, combinations of LSA and WordNet models were not investigated. But such a combination could still be useful for other tasks like information retrieval or word-sense disambiguation.

7.3 An afterthought

At the end of this thesis I would like to add some thoughts concerning the availability of resources. For me it was possible to undertake this work because I had access to the speech recognizer at ICSI.

Large vocabulary speech recognition research is focused on large systems. One can only do the research with a system at hand. This means also that there is a good part of engineering work in this type of research, which is also illustrated in the appendix. This resembles the situation in fundamental physics, where one may need a particle accelerator to run specific experiments. The dependence on large systems makes it difficult for some people to participate in this research.

I think that it would be beneficial for the community, if acoustic models and language models trained on large amounts of data would be provided to the public. In this way

a common database would be created for research, and it would be possible for more groups of researchers to participate in this interesting field.

A LSA modeling toolkit

This chapter describes the *Latent Semantic Analysis* (LSA) language modeling toolkit, that is implemented in `C` and is available from the website `http://userver.ftw.at/~pucher/resources.htm`.

A.1 Installation

Change the variable `TCLLIB=/usr/lib/libtcl` in `../src/Makefile` to point to your tcl library. Go to `../src` and type make. This will create the programs `lsalm`, `trainmodel` and `ngram-count`. In this version only the `i686` architecture is supported. For other architectures (`solaris`) get the *Stanford Research Institute Language Modeling* (SRILM) (Stolcke, 2002) libraries.

A.2 Usage

A.2.1 Training

To train an LSA model use:

`trainmodel`	`-data`	`train.data`	*Data file containing document boundaries*
	`-vocab`	`train.vocab`	*Vocabulary file. Each word in the vocabulary must appear in the data file*
	`-lap2`	`lap2file`	*Lap2 file containing parameters for SVD. The two integer values may not be bigger than the number of documents.*
	`-docbound`	`dbound`	*The document boundary string*
	`-output`	`lsa-train`	*Basename for output files*

To see all parameters of trainmodel call `trainmodel -help`. To train an n-gram model use the `ngram-count` program documented on the SRILM website (`http://www.speech.sri.com/projects/srilm/`).

A.2.2 Testing

To test an LSA model use:

To test an n-gram model skip the first three parameter of the `lsalm` program in the above example. An introduction to `ngram` and `ngram-count` is found in Stolcke

`lsalm`	`-lsamodel`	`lsa.term.entrop`	*Term entropy lsa model file*
	`-interpolate`	`infg`	*Interpolation method*
	`-docbound`	`dbound`	*The document boundary string set in quotes. If not set sentence boundary is used.*
	`-ppl`	`evalfile`	*File to compute perplexities*
	`-lm`	`train-ngram.gz`	*Language model*
	`-order`	`4`	*n-gram model order*
	`-debug`	`2 > lsa-infg-evalfile`	*Debug and output*

(2002). To see all parameters of `lsalm` call `lsalm -help`. The following parameters are `lsalm`-specific.

`-lsamodel:`	*term-entropy lsa model file*
`-modelname:`	*modelname for html output*
`-lambdangram:`	*ngram model weight for linear, log-linear and infg interpolation.*
`-lambdalsa:`	*lsamodel weight for linear, log-linear and infg interpolation.*
`-binmodel:`	*binmodel file*
`-nonorm:`	*do not use normalization for loglin and infg interpolation*
`-1besthist:`	*add 1-best of last utterance to pseudodoc history, for rescoring*
`-lsaslice:`	*maximum probability part for lsa models for infg interpolation*
`-nosqrts:`	*do not use square root of Singular Values*
`-initsent:`	*init pseudodoc at beginning of sentence*
`-html:`	*output html comparison file use with -debug 2*
`-docbound:`	*document boundary*
`-decay:`	*weight for lambda decay in (0,1]*
`-exp:`	*exp for similarity smoothing*
`-interpolate:`	*interpolation method for lsa and n-gram [infg, loglin]*
`-zentropmin:`	*replace zero entropies by this value*
`-skipzentrop:`	*delete zero entropy words from lsa model*
`-trainlambda:`	*does training of lambda values for lsa-ngram interpolation*
`-doccluster:`	*document cluster center file*
`-dumpfeat:`	*dump similarity features, needs doccluster*
`-mindumpsim:`	*minimum similarity for counting as feature*

A.2.3 Data format

Term.entrop file format

4	2			*words*	*singval*		
0.0022	0.0023			*singval1*	*singval2*		
i	0.9384	0.0000	-0.0001	*word*	*entropy*	*wvecval1*	*wvecval2*
think	0.0000	0.0000	-0.0000	-	-	-	-
this	0.9384	0.0000	-0.0001	-	-	-	-
works	0.0000	0.0000	-0.0000	-	-	-	-

Binmodel file format

3		*bins*	
-0.184038	2.07123e-06	*end of similarity interval*	*bin probability*
0.234566	3,40712e-06	-	-
0.999516	4.07123e-02	-	-

If the similarity is bigger than the last interval boundary, the last intervals bin probability is used.

Doc-cluster file format

3	2	*cluster centers*	*docvecvalues*
0.000003	-0.000142	*docvecvalue1*	*docvecvalue2*
0.000003	-0.000002	-	-
0.000003	-0.000142	-	-

A.2.4 HTML output

When `lsalm` is used with the `-html` option it produces *HyperText Markup Language* (HTML) output of the following style:

```
  we did not talk about pipes
p( we | ) = [2gram] [0.0115263] 0.0131275 [ -1.88182 ]
meet-th:  0.005114 fisher:  0.008618 web:  0.025500
p( did | we ...)  = [3gram] [0.0129977] 0.0147011 [ -1.83265 ]
meet-th:  0.007520 fisher:  0.009937 web:  0.035061
p( not | did ...)  = [4gram] [0.108605] 0.103255 [ -0.986088 ]
meet-th:  0.004505 fisher:  0.004057 web:  0.018205
p( talk | not ...)  = [3gram] [0.000583654] 0.000588433 [ -3.2303 ]
meet-th:  0.014041 fisher:  0.016797 web:  0.039907
p( about | talk ...)  = [3gram] [0.299694] 0.267013 [ -0.573467 ]
meet-th:  0.005322 fisher:  0.004451 web:  0.021566
p( pipes | about ...)  = [2gram] [4.84618e-06] 4.07627e-06 [ -5.38974 ]
```

```
fisher:  0.168330 web:  0.122102
p( | pipes ...)  = [2gram] [0.0664626] 0.0723812 [ -1.14037 ]
1 sentences, 6 words, 0 OOVs 0 zeroprobs, logprob= -15.0344 ppl= 140.532
ppl1= 320.435
```

Light green means that the LSA model is better than the n-gram (very good), dark green means that the LSA model is better, but the word already appears in the context, e.g. it is a cache model effect (good).

If the whole document is taken as the context (not `-initsent`) then there are a lot of repetitions although words that are far away will have no effect due to the `-decay`.

Dark red means that the LSA model is worse than the n-gram (bad). Light red means that the LSA model is worse than the n-gram and that the word already appears in the context (very bad).

`meet-th:  0.005114 fisher:  0.008618 web:  0.025500` are the `lambdas` (for `infg` interpolation) for the different models named *meet-th*, *fisher* and *web*, respectively.

Bibliography

Bacon, F. (2002). *The New Organon*. Cambridge University Press.

Bahl, L. R., Jelinek, F., and Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **5**(2), 179–190.

Baker, J. K. (1975). The DRAGON system - An overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **23**(1), 24–29.

Banerjee, S. and Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th Int. Joint Conf. on Artificial Intelligence*, pages 805–810, Acapulco, Mexico.

Bellegarda, J. (2000a). Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, **88**(8), 1279–1296.

Bellegarda, J. (2000b). Large vocabulary speech recognition with multispan statistical language models. *IEEE Transactions on Speech and Audio Processing*, **8**(1), 76–84.

Bellegarda, J. (2004). Statistical language model adaptation: Review and perspectives. *Speech Communication*, **42**(1), 93–108.

Berry, M. (1993). SVDPACKC. Technical Report CS-93-194, University of Tennessee, Knoxville, TN, USA.

Birkhoff, G. and von Neumann, J. (1936). The logic of quantum mechanics. *Annals of mathematics*, **37**, 823–843.

Bos, J. and Markert, K. (2005). Combining shallow and deep NLP methods for recognizing textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 65–68, Southampton, UK.

Brill, E., Florian, R., Henderson, J. C., and Mangu, L. (1998). Beyond N-grams: Can linguistic sophistication improve language modeling? In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL '98)*, pages 186–190, Quebec, Canada.

Brown, P., DeSouza, P., Mercer, R., Della Pietra, V., and Lai, J. (1992). Class-based N-gram models of natural language. *Computational Linguistics*, **18**(4), 467–479.

Budanitsky, A. (1999). Lexical semantic relatedness and its application in natural language processing. Technical Report CSRG-390, Computer Systems Research Group, University of Toronto, Canada.

Budanitsky, A. and Hirst, G. (2001). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, 2nd Meeting of the North American Chapter of the ACL*, pages 29–34, Pittsburgh, USA.

Bulyko, I., Ostendorf, M., and Stolcke, A. (2003). Class-dependent interpolation for estimating language models from multiple text sources. Technical Report UWEETR-2003-0000, University of Washington, EE Department, Washington, USA.

Burger, S., MacLaren, V., and Waibel, A. (2004). ISL meeting transcripts part 1. In *Linguistic Data Consortium*, Philadelphia.

Carnap, R. (1956). *Meaning and Necessity*. Chicago University Press.

Carnap, R. and Bar-Hillel, Y. (1952). An outline of a theory of semantic information. Technical Report 247, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

Charniak, E. and Johnson, M. (2001). Edit detection and parsing for transcribed speech. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'01)*, pages 118–126, Pittsburgh, PA, USA.

Chelba, C. and Jelinek, F. (1998). Exploiting syntactic structure for language modeling. In *Proceedings of the Conference of the Association for Computational Linguistics (COLING-ACL'98)*, pages 225–231, Montreal, Canada.

Chelba, C., Engle, D., Jelinek, F., Jimenez, V. M., Khudanpur, S., Mangu, L., Printz, H., Ristad, E., Rosenfeld, R., Stolcke, A., and Wu, D. (1997). Structure and performance of a dependency language model. In *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH'97)*, volume 5, pages 2775–2778, Rhodes, Greece.

Chen, S. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, Massachusetts.

Chomsky, N. (1957). *Syntactic Structures*. The Hague.

Chomsky, N. (1969). Quines empirical assumptions. In D. Davidson and J. Hintikka, editors, *Words and Objections: Essays on the Work of W.V. Quine*, pages 53–68. Reidel, Dordrecht.

Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origin, and Use*. Praeger.

Cieri, C., Miller, D., and Walker, K. (2004). The Fisher corpus : A resource for the next generations of speech-to-text. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 69–71, Lisbon, Protugal.

Coccaro, N. and Jurafsky, D. (1998). Towards better integration of semantic predictors in statistical language modeling. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'98)*, pages 2403–2406, Sydney, Australia.

Cohen, M. H., Giangola, J. P., and Balogh, J., editors (2004). *Voice User Interface Design*. Addison-Wesley.

Cole, R., Mariani, J., Uszkoreit, H., Zaenen, A., and Zue, V., editors (1998). *Survey of the State of the Art in Human Language Technology*. Cambridge University Press.

Cook, S. A. (1971). The complexity of theorem-proving procedures. In *Proceedings of the 3rd annual ACM symposium on Theory of computing*, pages 151–158, Ohio, USA.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley & Sons.

Cowell, R. (1998). Introduction to inference for Bayesian networks. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 9–27. MIT Press.

Daniels, D. (1992). *Duchamp und die anderen*. DuMont.

Deerwester, S., Dumais, S., Furnas, G., and Landauer, T. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, **41**, 391–407.

Della Pietra, S., Della Pietra, V., Mercer, R., and Roukos, S. (1992). Adaptive language modeling using minimum discriminant estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'92)*, volume 1, pages 633–636, San Francisco, CA, USA.

Demetriou, G., Atwell, E., and Souter, C. (1997). Large-scale lexical semantics for speech recognition support. In *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH'97)*, pages 2755–2758, Rhodes, Greece.

Demetriou, G., Atwell, E., and Souter, C. (2000). Using lexical semantic knowledge from machine readable dictionaries for domain independent language modelling. In *Proc. of LREC 2000, 2nd International Conference on Language Resources and Evaluation*, pages 777–782, Athens, Greece.

Dempster, A., Laird, N., and D.B., R. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **39**, 1–38.

Deng, Y. and Khudanpur, S. (2003). Latent semantic information in maximum entropy language models for conversational speech recognition. In *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL'03)*, pages 56–63, Edmonton, Canada.

Descartes (1996). *Meditations on First Philosophy*. Cambridge University Press.

Fellbaum, C., editor (1998). *WordNet: An electronical lexical database*. MIT Press.

Fiscus, J. G., Radde, N., Garofolo, J. S., Le, A., Ajot, J., and Laprun, C. (2005). The rich transcription 2005 spring meeting recognition evaluation. In *Rich Transcription 2005 Spring Meeting Recognition Evaluation Workshop*, pages 369–389, Edinburgh, UK.

Garofolo, J. S., Laprun, C. D., Michel, M., Stanford, V. M., and Tabassi, E. (2004). The NIST meeting room pilot corpus. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Protugal.

Gettier, E. (1963). Is justified true belief knowledge? *Analysis*, **23**, 121–123.

Gildea, D. and Hofmann, T. (1999). Topic-based language models using EM. In *Proceedings of 6th European Conference On Speech Communication and Technology (Eurospeech'99)*, pages 2167–2170, Budapest, Hungary.

Gillick, L. and Cox, S. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'89)*, pages 532–535, Glasgow, UK.

Golub, G. and Van Loan, C. (1989). *Matrix Computations*. Johns Hopkins.

Grice, H. P. (1981). Presupposition and conversational implicature. In P. Cole, editor, *Radical pragmatics*, pages 183–98. Academic Press.

Heeman, P. (1998). POS tagging versus classes in language modeling. In *Proceedings of the 6th Workshop on Very Large Corpora*, pages 179–187, Montreal, Canada.

Hirst, G. and St-Onge, D. (1998). Lexical chains as representations of context for the deduction and correction of malapropisms. In *WordNet: An electronical lexical database*, pages 305–332. MIT Press.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI'99)*, pages 289–296, San Francisco, CA, USA.

Hunt, M. (1988). Evaluating the performance of connected word speech recognition systems. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'88)*, pages 457–460, New York, USA.

Ishiguro, H. (1990). *Leibniz's Philosophy of Logic and Language*. Cambridge University Press.

Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., and Wooters, C. (2003). The ICSI meeting corpus. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing, (ICASSP'03)*, pages 364–367, Hong Kong.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, **64**(4), 532–557.

Jelinek, F. (1990). Self-organized language modeling for speech recognition. In A. Waibel and K. F. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Kaufmann, San Mateo, CA.

Ji, G. and Bilmes, J. (2004). Multi-speaker language modeling. In *Proceedings of HLT-NAACL Conference*, pages 137–140, Boston, MA, USA.

Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Research in Computational Linguistics (ROCLING)*, pages 19–33, Tapei, Taiwan.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing*. Prentice Hall.

Kintsch, W. (2001). Predication. *Cognitive Science*, **25**, 173–202.

Klakow, D. (1998). Log-linear interpolation of language models. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*, pages 1695–1699, Sydney, Australia.

Kneser, R. and Ney, H. (1995). Improved backing-off for M-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95)*, volume 1, pages 181–184, Detroit, MI, USA.

Koumpis, K. and Renals, S. (2005). Content-based access to spoken audio. *IEEE Signal Processing Magazine*, **22**(5), 61–69.

Kozima, H. and Ito, A. (1995). Context-sensitive measurement of word distance by adaptive scaling of a semantic space. In *Proc. of the International Conference 'Recent Advances in Natural Language Processing', RANLP-95*, pages 161–168, Bulgaria.

Kripke, S. (1980). *Naming and Necessity*. Harvard University Press.

Kuhn, R. and De Mori, R. (1990). A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(6), 570–583.

Landauer, T. and Dumais, S. (1997). A solution to Plato's problem: The LSA theory of acquisition, induction and representation of knowledge. *Psychological Review*, **104**, 211–240.

Leacock, C. and Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. In *WordNet: An electronic lexical database*, pages 265–283. MIT Press.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26, Toronto, Canada.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 64–71, Madrid, Spain.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304, Madison, Wisconsin, USA.

Liu, Y., Shriberg, E., and Stolcke, A. (2003). Automatic disfluency identification in conversational speech using multiple knowledge sources. In *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH-INTERSPEECH'03)*, pages 957–960, Geneva, Switzerland.

Lukasiewicz, J. (1935). Zur Geschichte der Aussagenlogik. *Erkenntnis*, **5**, 111–131.

Lyons, J. (1995). *Linguistic Semantics. An Introduction*. Cambridge University Press.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, CA, USA.

Mangu, L., Brill, E., and Stolcke, A. (1999). Finding consensus among words: Lattice-based word error minimization. In *Proceedings of the 6th European Conference on*

Speech Communication and Technology (EUROSPEECH'99), volume 1, pages 495–498, Budapest, Hungary.

Manning, D., C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Markie, P. (Summer 2006). Rationalism vs. empiricism. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. `http://plato.stanford.edu/archives/sum2006/entries/rationalism-empiricism/`.

Miller, G. A. (1998). Nouns in WordNet. In *WordNet: An electronic lexical database*, pages 23–46. MIT Press.

Morato, J., Miguel Ángel, M., Lloréns, J., and Moreiro, J. (2004). Wordnet applications. In *Proceedings of the 2nd Global Wordnet Conference (GWC'04)*, pages 270–278, Brno, Czech Republic.

Morgan, N., Baron, D., Bhagat, S., Carvey, H., Dhillon, R., Edwards, J., Gelbart, D., Janin, A., Krupski, A., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., and Wooters, C. (2003). Meetings about meetings: Research at ICSI on speech in multiparty meetings. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, pages 740–743, Hong Kong.

Murray, G., Renals, S., and Carletta, J. (2005). Extractive summarization of meeting recordings. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH-INTERSPEECH'05)*, pages 593–596, Lisbon, Portugal.

Murveit, H., Butzberger, J., Digalakis, V., and Weintraub, M. (1993). Large-vocabulary dictation using SRI's DECIPHER speech recognition system: Progressive search techniques. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'93)*, volume 2, pages 319–322, Minneapolis, USA.

Neal, R. M. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–371. MIT Press.

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). WordNet::Similarity - Measuring the Relatedness of Concepts. In *Proc. of 5th Annual Meeting of the North American Chapter of the ACL (NAACL-04)*, pages 38–41, Boston, MA, USA.

Pedersen, T., Banerjee, S., and Patwardhan, S. (2005). Maximizing semantic relatedness to perform word sense disambiguation. Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute.

Pollard, C. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Pucher, M. (2005). Performance evaluation of WordNet-based semantic relatedness measures for word prediction in conversational speech. In *Proceedings of 6th International Workshop on Computational Semantics (IWCS-6)*, pages 332–342, Tilburg, Netherlands.

Pucher, M. and Huang, Y. (2005). Latent semantic analysis based language models for meetings. In *MLMI05, 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Edinburgh, UK.

Pucher, M., Huang, Y., and Çetin, Ö. (2006a). Combination of latent semantic analysis based language models for meeting recognition. In *Computational Intelligence 2006, Special Session on "Natural Language Processing for Real Life Applications", San Francisco, USA*, pages 465–469, San Francisco, USA.

Pucher, M., Huang, Y., and Çetin, Ö. (2006b). Optimization of latent semantic analysis based language model interpolation for meeting recognition. In *5th Slovenian and 1st International Language Technologies Conference*, pages 74–78, Ljubljana, Slovenia.

Putnam, H. (1976). The logic of quantum mechanics. In *Mathematics, Matter and Method*, pages 174–197. Cambridge University Press.

Quasthoff, U. and Wolff, C. (2002). The poisson collocation measure and its applications. In *Proceedings of the 2nd International Workshop on Computational Approaches to Collocations*, Vienna, Austria.

Quine, W. V. O. (1953a). On what there is. In *From a logical point of View*, pages 1–19. Harvard University Press.

Quine, W. V. O. (1953b). Two dogmas of empiricism. In *From a logical point of View*, pages 20–46. Harvard University Press.

Quine, W. V. O. (1960). *Word and Object*. MIT Press.

Quine, W. V. O. (1981). *Mathematical Logic*. Harvard University Press.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 448–453, Montreal, Canada.

Rosenfeld, R. (1994). Adaptive statistical language modeling: A maximum entropy approach. Technical Report CMU-CS-94-138, Carnegie Mellon University, Pittsburgh, PA, USA.

Rosenfeld, R. (1996). A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, **10**(3), 187–228.

Sanouillet, M. and Peterson, E., editors (1973). *The writings of Marcel Duchamp*. Oxford University Press.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Schofield, E. (2006). *Fitting maximum entropy models on large sample spaces*. Ph.D. thesis, Imperial College London, London, UK.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, **24**, 97–124.

Schwartz, R. and Chow, Y.-L. (1990). The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP*, volume 1, pages 81–84, Albuquerque, USA.

Shriberg, E. (2005). Spontaneous speech: How people really talk and why engineers should care. In *Proc. of Eurospeech-05*, pages 1781–1784, Lisboa, Portugal.

Solan, Z., Horn, D., Ruppin, E., and Edelman, S. (2005). Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(33), 11629–11634.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'02)*, volume 2, pages 901–904, Denver, Colorado, USA.

Stolcke, A., König, Y., and Weintraub, M. (1997). Explicit word error minimization in N-best list rescoring. In *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH'97)*, volume 1, pages 163–166, Rhodes, Greece.

Stolcke, A., Anguera, X., Boakye, K., Cetin, O., Grezl, F., Janin, A., Mandal, A., Peskin, B., Wooters, C., and Zheng, J. (2005). Further progress in meeting recognition: The ICSI-SRI spring 2005 speech-to-text evaluation system. In *Proceedings of NIST MLMI Meeting Recognition Workshop*, pages 463–475, Edinburgh, UK.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, **13**(2), 260–269.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM (JACM)*, **21**(1), 168–173.

Wandmacher, T. (2005). On the semantics of latent semantic analysis - A contrastive study. In *Proceedings of the Ninth ESSLLI Student Session*, Edinburgh, UK.

Wang, Y.-Y., Acero, A., and Chelba, C. (2003). Is word error rate a good indicator for spoken language? In *Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU'03)*, pages 577–582, St. Thomas, Virgin Islands, USA.

Wang, Y.-Y., Deng, L., and Acero, A. (2005). Spoken language understandng: An introduction to the statistical framework. *IEEE Signal Processing Magazine*, **22**(5), 16–32.

Widdows, D. and Stanley, P. (2003). Word vectors and quantum logic: Experiments with negation and disjunction. In *Proceedings of 8th Mathematics of Language Conference*, pages 141–154, Bloomington, Indiana, USA.

Wittgenstein, L. (1984). *Philosophische Untersuchungen*. Suhrkamp Verlag.

Wittgenstein, L. (2001). *Philosophical Investigations*. Blackwell Publishing.

Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, Mexico.

Zimmermann, M., Liu, Y., Shriberg, E., and Stolcke, A. (2005). A* based joint segmentation and classification of dialog acts in multi-party meetings. In *Proceedings of the 2005 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'05)*, pages 215–219, San Juan, Puerto Rico.