# IMPROVING N-GRAM PROBABILITY ESTIMATES BY COMPOUND-HEAD CLUSTERING

Joris Pelemans<sup>1</sup>, Kris Demuynck<sup>2</sup>, Hugo Van hamme<sup>1</sup>, Patrick Wambacq<sup>1</sup>

<sup>1</sup>Dept. ESAT, Katholieke Universiteit Leuven, Belgium <sup>2</sup>ELIS, Ghent University, Belgium

{joris.pelemans,hugo.vanhamme,patrick.wambacg}@esat.kuleuven.be kris.demuynck@elis.ugent.be

# ABSTRACT

Compounding is one of the most productive word formation processes in many languages and is therefore a main source of data sparsity in language modeling. Many solutions have been suggested to model compound words, most of which break the compound into its constituents and train a new model with them. In earlier work, we argued that this approach is suboptimal and we presented a novel technique that clusters new, domain-specific compound words together with their semantic heads. The clusters were then used to build a class-based n-gram model that enabled a reliable estimation of n-gram probabilities, without the need for additional training data. In this paper, we investigate how this "semantic head mapping" can best be made an integral part of the language modeling strategy and find that, with some adaptations, our technique is capable of producing more accurate compound probability estimates than a baseline word-based n-gram language model, which lead to a significant word error rate reduction for Dutch read speech.

Index Terms- LVCSR, language models, n-grams, data sparsity, word clusters

# 1. INTRODUCTION

Language data sparsity has been addressed in different ways throughout the history of automatic speech recognition (ASR). In the early days of n-gram language models (LMs), a series of smoothing techniques were proposed that redistribute the available probability mass and put aside some of the mass for unseen events [1, 2, 3, 4]. A more versatile approach was suggested by Brown et al. [5] who assign words to classes, each word in a class having similar properties. Instead of word n-gram probabilities, class n-gram probabilities are calculated to achieve a higher level of abstraction.

Recently the focus has shifted to LMs with continuous distributions in the form of semantic-analytical and neural network-based approaches. Topic models such as LSA [6], pLSA [7] and LDA [8] are bag-of-words models that are inspired by the notion of word co-occurrence and can be combined with n-grams to model both local and long-distance phenomena. The current state-of-the-art in language modeling however is based on neural networks (NNs). Amongst others, [9] and [10] have shown that NNs are capable of integrating both syntactic and semantic properties, but unfortunately these models are currently too expensive for direct integration into an ASR decoder.

Although this shows that there has been great progress in language modeling, all of the above models rely on corpus statistics and therefore, none of them are very good at estimating reliable probabilities for new words or words that occur infrequently. Unfortunately

(from a language modeling point of view), new words will continue to be created, perhaps even at a faster pace than before: although the rapid expansion of the Internet enables the gathering of Big Data, it goes hand in hand with numerous new words e.g. Netizen, tweeple, infobesity, ...

New terms may be coined in various ways. In many languages, compounding is one of the most productive processes which induces the frequent creation of numerous new words all over the world. This gives rise to challenging problems in speech and language research which has been addressed by several authors for languages as diverse as German [11], Mandarin [12], and Hindi [13].

In recent work [14] we proposed an alternative to the popular, yet suboptimal decompound-recompound approach. By mapping compounds onto their semantic heads, our technique effectively clusters words regardless of their frequency. The word clusters can be used to build a class-based n-gram LM enabling a more reliable estimation of n-gram probabilities for compounds. This proof of concept showed that semantic head mapping is a viable technique to add new compounds to a given LM, which is a typical use case in domain adaptation.

In this paper, we investigate how semantic head mapping can best be made an integral part of the language modeling strategy. In other words, we would like to answer the following research question: given some training data, what is the best way to model compounds in an n-gram LM?

The rest of the paper is organized as follows: Section 2 briefly recapitulates our earlier work on semantic head mapping. In Section 3, we highlight an improvement on our existing work in domain adaptation. We then discuss in Section 4 how our framework needs to be altered in order to apply it in a more general fashion and validate this approach experimentally in Section 5. We end with a conclusion and a description of future work.

# 2. SEMANTIC HEAD MAPPING

Semantic head mapping (SHM) is a technique that tackles data sparsity in n-gram LMs by building a class-based model with compoundhead clusters. In this section, we briefly review the different steps that are involved in this process. For a more detailed explanation, we refer to [14].

## 2.1. Idea

Compounding is the process of word formation which combines two or more lexemes into a new lexeme e.g. energy+drink. Most compounds are endocentric i.e. they consist of a semantic head and modifiers which introduce a hyponym-hypernym or type-of relation. The position of the head varies among languages and often corresponds to a specific manner of constituent combination. In what follows we will focus on compounds in Dutch, which is our native language and the target language in our experiments, but we believe that the presented ideas extend well to other languages on the condition that, like Dutch, they have a lexical morphology with concatenative and right-headed compounding, which at least includes most Germanic languages.

In contrast to most other types of word formation, compounding is a very transparant way of creating new words: the meaning of the new word can be immediately understood by examining its constituents. This is a property that has not been (fully) exploited in previous literature. In [14] we argued that for most compound words, the head has the unique property of carrying inherent class information, which is obviously the case for the predominant class of endocentric compounds. By mapping a compound onto its semantic head we effectively apply a clustering that does not depend on any external information and can hence be applied to all compounds, regardless of their frequency in a training corpus. The clustering can be used to build a class-based n-gram model, where compounds are clustered together with their heads.

### 2.2. Finding the semantic heads

The semantic heads are obtained via a two-step process: (1) a generation module which generates all possible decompounding hypotheses; and (2) a selection module which selects the most plausible head. The generation module is a brute-force (recursive) lexicon lookup for all possible substrings of the compound. Constraints can be set on the lengths  $L_h$  and  $L_m$  of the substrings and on the lexicon sizes  $V_h$  and  $V_m$  for heads and modifiers respectively. In the selection module we prune the unlikely hypotheses and select the most likely head candidate. Previous experiments show that the trivial criterion of maximum head length outperforms criteria based on corpus statistics [14].

### 2.3. LM incorporation

Semantic head mapping is inspired by class-based n-gram models [5]. The idea of class n-grams is that words are similar to others in their meaning and syntactic function. Grouping such words into classes can help overcome the data sparsity in training material, since the prediction of infrequent or unseen words is then based on the behavior of similar words that have been seen (more often). Formula 1 shows how the n-gram probabilities are calculated:

$$P(w_k|w_1^{k-1}) = P(C_k|C_1^{k-1})P(w_k|C_k)$$
(1)

where  $w_k$  and  $C_k$  denote the word and class at position k respectively and  $w_1^{k-1}$  and  $C_1^{k-1}$  denote the word and class sequences from positions 1 to k - 1.

A problem with class-based approaches is that they tend to overgeneralize: the hypothesis that all words in the same class behave in a similar fashion is too strong. Moreover, clustering words into appropriate classes is not an easy problem, especially for rare words which are typically not included in a taxonomy and appear too infrequently for corpus-based clustering techniques.

Our approach essentially consists of building a class-based ngram model, where classes are created, using the SHM clustering scheme discussed above. As opposed to most clustering algorithms, SHM does not just shift the data sparsity problem from the LM to the clustering: each (endocentric) compound has a semantic head and can thus be clustered, whether it be frequent or infrequent. In addition, since most compounds are in fact subtypes of their semantic head, overgeneralization is limited.

In [14] we applied SHM in the context of domain adaptation, where we only integrated compound-head clusters into the LM for unseen words. Since the compound is well represented by its semantic head, we used the n-gram probability of the head as the class n-gram probability  $P(C_k|C_1^{k-1})$  for each member. As no count information was available, the within-class probability was estimated by assigning a frequency count  $\hat{c}(w)$  to each of the unseen compounds w and normalizing by the count of all members of the class  $C_{head}$ , defined by the semantic head.

## 3. IMPROVED CLUSTERING: WORD-AFFIX DISAMBIGUATION

Dutch has a number of letter sequences that can act both as a word and as an affix, most of which are relatively short. The SHM system we described in [14] cannot distinguish between the two, since it has no access to semantic knowledge. This leads to a number of errors where a prefix is interpreted as a noun modifier e.g. ver+lengen or where a suffix is interpreted as the semantic head e.g. gevaar+lijk. Although in many cases this phenomenon can be prevented by increasing the minimal length of the modifiers/heads, this also lowers the system's recall. Better solutions are at hand, without the need for semantic ontologies. To address this issue, we made lists of all the sequences of k letters at the start and end of each word in our corpus and counted how many times they occurred as a word or an affix. The idea is that affixes perform a syntactic role that can be applied to a large number of words and are therefore more likely to occur than regular words. Letter sequences that occur significantly less as a word than as an affix, are then assumed to be prefixes/suffixes and are subsequently removed from the modifier/head lexica in the generation module. This solution also has the positive side effect that some spelling mistakes and foreign heads are filtered out. In what follows, the extent of affix-word ambiguity handling is indicated by the amount of modifiers  $a_m$  and heads  $a_h$  that were removed from the constituent lexica.

## 4. ADAPTATIONS FOR IN-VOCABULARY MAPPING

Semantic head mapping needn't be restricted to the context of domain adaptation: even for compounds for which training material exists, the clustering can be beneficial. In this section, we will show how our existing system can be adapted for in-vocabulary mapping.

#### 4.1. Conservative mapping

In domain adaptation, only unseen words undergo the process of SHM. As such, mapping errors are not likely to be costly: if a noncompound is mapped, or if a compound is mapped to the wrong head, it is unlikely that this will do much damage to the LM. The unseen word will most likely not be recognized, which was already the case, even before SHM; and the probability mass of the head will only be reduced by a small amount.

In the case of the more general approach however, incorrect mappings may be catastrophical. Because the amount of words that undergo SHM is much larger, the probability mass of the hypothesized head may be severely reduced. Moreover, many words that were correctly recognized before, may no longer be recognized. Clearly, in this scenario, SHM should be undertaken with more caution. This caution was implemented by attributing a larger relative importance to SHM precision than to recall during model optimization. We achieved this by using the  $F_{\beta}$ -score which is well known in information retrieval:

$$F_{\beta} = (1 + \beta^2) \cdot \frac{precision.recall}{(\beta^2.precision) + recall}$$
(2)

where  $\beta$  indicates the weighting ratio of recall and precision. In our case,  $\beta$  should thus be lower than 1.

## 4.2. Morphological knowledge

In our earlier work [14], we concluded that the use of part-of-speech (POS) information did not yield any improvement to the semantic head mapping of out-of-vocabulary words. This was mostly caused by incorrect POS tags assigned to both the infrequent compounds and to a lesser extent also the constituents. The data sparsity problem had thus shifted from language modeling to POS tagging. In our current setup, where we target more frequent compounds, this is not the case. The semantic head mapper is less hampered by sparsity and is helped by the inclusion of sophisticated linguistic knowledge based on POS rules.

As a first rule we force the compound and its head to have identical POS tags, as a change in POS tag is unlikely to correspond to a correct semantic head. This compound-head POS constraint requires the POS tagger to be fine-grained in order to distinguish between e.g. noun genders or verb tenses. To further constrain the search space we apply compounding rules that only allow combining constituents with certain POS tags e.g. concatenating two nouns is productive, hence allowed, but concatenating two verbs is not. These rules are implemented based on [15] where the author showed that they severely restrict overgeneration.

As a final SHM improvement we investigate the effect of a lexical database that contains hand-made morphological analyses of a large amount of words, which we expect to be beneficial for both recall and precision.

### 4.3. Data-driven LM integration

In the context of domain adaptation we did not have any training data for the compounds to be mapped. As was mentioned in Section 2.3, we therefore had to resort to several estimation techniques for the class n-gram and within-class probabilities respectively. Here, however, we do have access to training material, which facilitates the estimation process.

For the estimation of the class n-gram probability we no longer only use the n-grams of the semantic head, but also include the ngrams of all the class members i.e. the compounds that were mapped to this head. For the within-class probability, we can simply plug in the counts of the compounds. For compounds that are too infrequent, we can still attempt empirical estimation.

### 5. EXPERIMENTS AND DISCUSSION

## 5.1. Data

Our LM training data consists of a collection of normalized newspaper texts from the Flemish digital press database Mediargus which contains 1104M word instances (tokens) and 5M unique words (types) from which we extracted all the mentioned vocabularies and word frequencies. Vocabularies of V words always contain the V most frequent words in Mediargus. They were converted into phonemic lexica using an updated version of [16] and integrated, together with the created LMs (see Section 5.3), into the recognizer described in [17]. The semantic head mapper is the same that was used in [14], except for the adaptations mentioned in the previous sections. The POS tags, used for the morphological rules, were generated by running the Dutch POS tagger Frog [18] on the entire LM training data, after which only the most frequent POS tags were kept for each word. The lexical database from which we extracted decompoundings is CELEX [19] which contains morphological analyses of 122k types of which 68k are compounds.

The ASR development and test data each consist of 200 fragments of the Flemish part of the Corpus Spoken Dutch [20] component o, which contains read speech. Both of them contain around 6.5h of speech and consist of about 60k tokens, produced by 10k types. As opposed to our earlier work, the data sets were not reduced to fragments that only contain compounds, to show that the SHM of in-vocabulary words does not yield worse LM statistics and hence WERs for non-compounds.

### 5.2. SHM improvements

Although the SHM system that was used in [14] did a fine job in handling unseen compounds, it would not be suited for large-scale application, because it has too many false positives. As we mentioned in 4.1, the damage done by an incorrect mapping is likely to be larger than the positive impact of a correct mapping, so only a system that attaches more importance to precision than to recall would be suited to our purposes. The F-score was derived so that  $F_{\beta}$  measures the accuracy of a system that attaches  $\beta$  times as much importance to recall as to precision.

We did not attempt a thorough optimization of  $\beta$ , as it is only an indirect measure of the fitness of our semantic head mapper. Instead, we performed ASR experiments with retrained class-based ngram LMs, using the optimal clustering systems wrt precision and recall and chose the one that performed best on the development data. Since ASR experiments are computationally expensive we then determined an optimal value of  $\beta$  for future experiments. Table 1 shows precision, recall, WER and F-scores on the development data for various systems. Note that the WERs are achieved by feeding the 700k least frequent words to the semantic head mapper (see Section 5.3).

In our initial attempts to reduce the number of false positives we found that increasing the minimum constituent length to 5 or 6, and pruning the most probable affixes, drastically decreased the recall and as such limited the potential of SHM<sub>1</sub> (P=88.5%, R=12.0%, WER=27.55%). This shows that more intelligent pruning is necessary and the POS rules as stated in [15] provide exactly this. The application of each rule separately shows that most of the compounds follow one of the following 2 rules:

- 1. noun + noun = noun (*spraak*+*herkenning*)
- 2. verb stem + noun = noun (*speel*+*plein*)

When combined, these rules allow for a precise SHM setup  $(SHM_2)$  with better recall and lower WER (P=90.6%, R=25.0%, WER=27.25%), even when allowing relatively short constituents of length 4. Although this setup disallows all other possible decompoundings and thus employs a very rigorous pruning, it still benefits from affix-word ambiguity resolution: pruning the 10,000 most probable suffixes yields an absolute precision increase of 2.7% at the cost of a recall decrease of 3.1%, which lead to a further WER reduction of 0.11% (SHM<sub>3</sub>).

In fact, when evaluated on the development set, this simple system turns out to be very competitive to systems that include external

	Vm	$V_{h}$	$L_{m}$	$\mathbf{L}_{\mathbf{h}}$	$\mathbf{a}_{\mathbf{m}}$	$\mathbf{a_h}$	rules	db	P	R	WER	$\mathbf{F}_{0.2}$	$F_{0.5}$
$SHM_1$	600k	200k	5	6	200	300	-	no	88.5	12.0	27.55	71.07	38.9
$SHM_2$	20k	20k	4	4	0	0	1-2	no	90.6	25.0	27.25	82.29	59.42
SHM <sub>3</sub>	20k	20k	4	4	0	10k	1-2	no	93.3	21.9	27.14	82.90	56.48
$SHM_4$	20k	20k	4	4	0	10k	1-2	yes	89.6	31.3	27.15	83.61	65.28
SHM <sub>5</sub>	200k	100k	3	4	0	0	1-5	yes	80.2	46.6	27.09	78.04	70.09

**Table 1.** SHM fitness as measured by precision (P), recall (R), WER (with 700k least frequent words fed to SHM) and F-scores on the development data. The columns indicate (fltr) the generation parameters ( $V_m$ - $a_h$ ), which POS rules were employed, and whether or not the system had access to a lexical database.

knowledge from a lexical database with hand-made morphological analyses. Only  $SHM_5$  is able to beat it by a small margin (P=80.2%, R=46.6%, WER=27.09%), employing these additional morphological rules:

- 3. adverb + noun = noun (*samen+stelling*)
- 4. noun + adj = adj (kurk + droog)
- 5. quantifier + quantifier = quantifier (*vijf+tien*)

Notice that  $SHM_4$  is a nice illustration of the improvements described in this paper. It has a precision that's comparable to (and even a bit higher than)  $SHM_1$ , but a recall that's almost 20% higher!

At first glance, it is not obvious what  $\beta$  one should employ for future experiments. However, if we distinguish between the setups that have access to the lexical database and those that don't, we find that, although both groups require emphasis on precision, the first group attributes more weight to recall ( $\beta \ge 0.5$ ) than the second group ( $\beta \le 0.2$ ). This is logical, since with access to external morphological knowledge a system can afford to make more mistakes.

### 5.3. In-vocabulary mapping

For a thorough investigation of the in-vocabulary mapping, we trained an open vocabulary word-based 3-gram LM with modified Kneser-Ney back-off on the 800k most frequent words in Mediargus. Different fractions of the lexicon were fed to the semantic head mapper to investigate to what extent SHM would be useful for (in)frequent compounds. Fractions were chosen based on word (in)frequency e.g. an indicated fraction of 50% means that the 400k least frequent words in the LM training data undergo decompounding. Only those that are judged to be a compound by the SHM system are clustered together with their semantic head. For each fraction then, various SHM settings were explored and optimized on the development data, as was already explained in Section 5.2. We found that the optimal system corresponds to SHM<sub>5</sub> in Table 1, which shows that 0.5 is a reasonable value for  $\beta$ , especially for systems that have access to external morphological knowledge.

Table 2 gives an overview of the WERs on both the development and evaluation data for this optimal system. It can be seen that for all fractions, SHM constitutes a small, but significant improvement over the baseline word-based n-gram LM, which is indicated by a 0% fraction. Another thing to remark, is that the results continue to improve as the fraction undergoing SHM becomes larger, unless when SHM is applied to the complete lexicon. We suspect that this is due to some overgeneralization by clustering more likely to have a unique meaning and context that is different from its head. If it is clustered together with the head and the potentially many compounds that have the same head, it will lose some of its probability mass to these other class members, which may be harmful for contexts that are unique to  $w_{freq}$ . It should however be noted that even in these extreme cases, SHM still outperforms the baseline.

		fraction of lexicon undergoing SHM									
	0%	50%	75%	87.5%	93.75%	100%					
	0	400k	600k	700k	750k	800k					
dev	27.70	27.19	27.12	27.09	27.08	27.31					
eval	26.68	26.35	26.35	26.31	26.27	26.43					

**Table 2**. WERs (in %) in function of fraction of the 800k lexicon undergoing SHM, compared to a baseline (0%) 3-gram LM. Fractions correspond to the least frequent words in the LM training data.

The WER reductions are considerable, given the fact that we are only trying to correct compound words. This becomes more clear when we check how many of the previously misrecognized compounds were actually corrected. In the scenario of feeding 93.75% or 750k words of the lexicon to the head mapper, approximately 220k words are mapped of which only 592 types and 632 tokens actually occur in the test data. Of these, 75.7% of the types and 76.3% of the tokens were correctly recognized after applying SHM to the 3gram LM. Other scenarios are similar. The fact that clustering more than a quarter of the complete vocabulary does not harm the WERs, but on the contrary, that after clustering, more than 75% of the compounds are correctly recognized, shows that the achieved improvements are substantial and indicates that semantic head mapping is a better way to model compounds in Dutch than classical word-based n-gram LMs.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we have described improvements to our semantic head mapping technique which enable its application on in-vocabulary words. By including word-affix ambiguity filters, morphological rules based on part-of-speech and a lexical database with morphological analyses we created high precision semantic head mappers with increased recall (e.g. the 20% increase for a system with 90% precision). We have proven empirically that the resulting compound-head clusters, when used in a class-based n-gram language model, are capable of estimating more reliable n-gram probabilities of seen compounds i.e. compounds for which training data is available. Moreover, we have shown that a more conservative, but accurate mapper may be applied to a large fraction of the lexicon to achieve a compound recognition of more than 75% and a significant WER reduction, compared to a baseline word-based n-gram language model.

We believe that the compound-head clusters are a useful source of information that don't have to be limited to class-based n-gram language models. In the future, we will investigate whether they can be successfully applied to more advanced LM techniques such as neural network language models.

## 7. REFERENCES

- I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 1085–1094, 1991.
- [2] I.J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, pp. 237–264, 1953.
- [3] Reinhard Kneser and Hermann Ney, "Improved backing-off for m-gram language modeling," in *Proc. ICASSP*, 1995, vol. I, pp. 181–184.
- [4] Stanley F. Chen and Joshua Goodman, "An empirical study of smoothing techniques for language modeling," Tech. Report TR-10-98, Computer Science Group, Harvard U., Cambridge, MA, August 1998.
- [5] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, pp. 467–479, 1992.
- [6] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
- [7] Thomas Hofmann, "Probabilistic latent semantic analysis," in *Proc. of Uncertainty in Artificial Intelligence*, 1999, pp. 289– 296.
- [8] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [9] Frederic Morin and Yoshua Bengio, "Hierarchical probabilistic neural network language model," in *AISTATS*, 2005, pp. 246– 252.
- [10] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010, pp. 1045–1048.
- [11] Markus Nußbaum-Thom, Amr El-Desoky Mousa, Ralf Schlüter, and Hermann Ney, "Compound word recombination for German LVCSR," in *Proc. Interspeech*, Florence, Italy, 2011, pp. 1449–1452.
- [12] Jie Zhou, Qin Shi, and Yong Qin, "Generating compound words with high order n-gram information in large vocabulary speech recognition systems," in *Proc. ICASSP*, 2011, pp. 5560–5563.
- [13] S. R. Deepa, Kalika Bali, Angarai Ganesan Ramakrishnan, and Partha Pratim Talukdar, "Automatic generation of compound word lexicon for Hindi speech synthesis," in *Proc. LREC*, 2004.
- [14] Joris Pelemans, Kris Demuynck, Hugo Van hamme, and Patrick Wambacq, "Coping with language data sparsity: Semantic head mapping of compound words," in *Proc. ICASSP*, 2014.
- [15] Vincent Vandeghinste, "Lexicon optimization: Maximizing lexical coverage in speech recognition through automated compounding," in *Proc. LREC*, 2002.
- [16] Kris Demuynck, Tom Laureys, and Steven Gillis, "Automatic generation of phonetic transcriptions for large speech corpora," in *Proc. ICSLP*, 2002, vol. I, pp. 333–336.

- [17] Kris Demuynck, Antti Puurula, Dirk Van Compernolle, and Patrick Wambacq, "The ESAT 2008 system for N-Best Dutch speech recognition benchmark," in *Proc. ASRU*, 2009, pp. 339–343.
- [18] Antal van den Bosch, B. Busser, S. Canisius, and Walter Daelemans, "An efficient memory-based morphosyntactic tagger and parser for dutch," in *Computational Linguistics in the Netherlands 2006: Selected papers from the seventeenth CLIN Meeting*, P. Dirix, Ed., Utrecht, 2007, LOT, pp. 191–206, LOT.
- [19] Harald Baayen, Ronald Piepenbrock, and Louis Gulikers, "The CELEX lexical database (release2) [CD-ROM]," Linguistic Data Consortium, Philadelphia, 1995.
- [20] N. Oostdijk, "The Spoken Dutch Corpus," *The ELRA Newsletter*, vol. 5, no. 2, pp. 4–8, 2000, http://lands.let.ru.nl/cgn/.