COPING WITH LANGUAGE DATA SPARSITY: SEMANTIC HEAD MAPPING OF COMPOUND WORDS

Joris Pelemans[1], *Kris Demuynck*[2], *Hugo Van hamme*[1], *Patrick Wambacq*[1]

[1]Dept. ESAT, Katholieke Universiteit Leuven, Belgium
[2]DSSP, ELIS, Ghent University, Belgium

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

ABSTRACT

In this paper we present a novel clustering technique for compound words. By mapping compounds onto their semantic heads, the technique is able to estimate n-gram probabilities for unseen compounds. We argue that compounds are well represented by their heads which allows the clustering of rare words and reduces the risk of overgeneralization. The semantic heads are obtained by a two-step process which consists of constituent generation and best head selection based on corpus statistics. Experiments on Dutch read speech show that our technique is capable of correctly identifying compounds and their semantic heads with a precision of 80.25% and a recall of 85.97%. A class-based language model with compound-head clusters achieves a significant reduction in both perplexity and WER.

Index Terms— n-grams, compounds, clustering, sparsity, OOV

1. INTRODUCTION

Although n-grams are still the most popular language model (LM) approach in automatic speech recognition (ASR), they have two apparent disadvantages: first of all, they only operate locally and hence cannot model long-span phenomena such as sentence or document wide semantic relations. This can be partly alleviated by combining n-grams with semantic-analytical techniques such as LSA [1], pLSA [2] and LDA [3], but continues to be a challenging research task. The second disadvantage is data sparsity: there is not enough training material to derive reliable statistics for every possible (spoken) word sequence of length n, especially when n is large. Many word sequences and even single words only occur a limited number of times in the training material while others don't occur at all. This led to a series of smoothing techniques that redistribute the probability mass and put aside some of the mass for unseen events [4, 5, 6, 7].

While improving results, smoothing doesn't solve the actual problem. A more versatile approach was suggested by Brown et al [8] 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 and reduce data sparsity. Although this approach seems very similar to the way humans view words, it introduces the new and far from trivial problem of clustering words into classes. Indeed, for the idea of class n-grams to work, the words in a class should be both semantically and syntactically similar. This is a challenging task and even if it is accomplished successfully it may still suffer from overgeneralization because of the many senses words can have [9]. In addition, most clustering algorithms rely either on a taxonomy or on corpus statistics, where rare words are often not represented (well enough).

In this paper we present a novel clustering technique for compound words. By mapping compounds onto their semantic heads, the technique is able to estimate n-gram probabilities for unseen compounds. We argue that compounds are well represented by their heads which allows the clustering of rare words and reduces the risk of overgeneralization. This approach is especially interesting for domain adaptation purposes, but can also be applied in more general contexts and research areas which rely on n-gram models such as machine translation and optical character recognition. The technique is evaluated on Dutch read speech, but the idea may extend to languages with similar compound formation rules.

The paper is organized as follows: section 2 gives a linguistic description of compounds and zooms in on compounding in Dutch. In section 3 we discuss related work. The remainder of the paper focuses on our new approach. Section 4 explains semantic head mapping (SHM) in more detail and section 5 handles the integration of the compound-head clusters into the LM. Finally, section 6 validates the merits of the technique experimentally. We end with a conclusion and a description of future work.

2. COMPOUND WORDS

2.1. Linguistic description

Compounding is the process of word formation which combines two or more lexemes into a new lexeme e.g. *energy+drink*. This should not be confused with derivation[1] where a lexeme is combined with an affix instead of another lexeme e.g. *recreation+al*. Compound formation rules vary widely across language types. This section is not meant to give an exhaustive overview, but rather to introduce the concepts relevant to our approach. Examples are limited to Germanic and Romance languages which are most familiar to the authors.

The manner in which compound constituents are combined differs from language to language. Some languages put the constituents after each other, which is (mostly) the case for English. Others apply concatenation, possibly with the insertion of a binding morpheme. Still others use prepositional phrases to describe a relation between the head and the modifier e.g. the Spanish *zumo de naranja (lit: juice of orange)* or the French *machine à laver (lit: machine to wash)*.

Compounds can be broadly classified into 4 groups, based on their constituent semantics:

[1]Compounding and derivation are not the only word formation processes, but they are by far the most productive.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

1. **endocentric compounds** consist of a semantic head and modifiers which introduce a hyponym-hypernym or type-of relation e.g. *energy drink*.
2. **copulative compounds** have two semantic heads, both of which contribute to the total meaning of the compound e.g. *sleepwalk*.
3. **appositional compounds** consist of two (contrary) classifying attributes e.g. *actor-director*.
4. **exocentric compounds** have a meaning that cannot be transparently derived from its constituent parts e.g. *skinhead*.

The position of the head also varies among languages and often corresponds to a specific manner of constituent combination. Germanic languages predominantly use concatenation with the semantic head taking the rightmost position in the compound. Romance languages on the other hand are typically left-headed, applying the prepositional scheme mentioned above on the right-hand side.

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 to other languages on the condition that, like Dutch, they have a lexical morphology with concatenative and right-headed compounding.

2.2. Dutch compounding

Like most Germanic languages, Dutch is a language with a relatively rich lexical morphology in the sense that new (compound) words can be made by concatenating two or more existing words e.g. *voor+deur+klink = voordeurklink (front door handle)*. Often the words are not simply concatenated, but separated by a binding morpheme which expresses a possessive relation between the constituents or facilitates the pronunciation or readability of the compound e.g. *tijd+s+druk = tijdsdruk (time pressure)*. The majority of compounds in Dutch are right-headed and endocentric; some are copulative or appositional and a minority is exocentric [10]. Left-headed compounds do occur e.g. *kabinet-Vandeurzen (cabinet [of minister] Vandeurzen)*, but are rare.

3. RELATED WORK

3.1. Decompound-recompound approaches

In many languages compounding is a productive process which induces the frequent creation of numerous new words all over the world. This process results in observing many compound words, most of them occurring rarely or with low frequency. As a consequence these words are not included in a n-gram LM or are included with a very unreliable probability. Moreover, even if sufficient training data is available, a typical application is limited in the number of words it can include in its vocabulary. These issues give 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].

The most popular approach to address compounds in Dutch (and also in other languages) is to split them into their constituent parts and add these to the lexicon and LM. After recognition, the constituents are then to be recombined. Earlier research based on rule-based [14] and data-driven decompounding [15, 16] has shown that this does indeed reduce the word error rate (WER) for Dutch ASR.

This technique was mainly developed to achieve maximal coverage with minimal vocabulary, and has several disadvantages wrt language modeling: (1) recompounding the emerging constituents is not trivial because many constituent pairs also exist as word pairs; (2) for unseen compounds, the constituents have never occurred together, resulting in the LM basing its decision on unigram probabilities; and (3) given that in Dutch compounds the first constituents generally play the role of modifiers while the last constituent acts as semantic head of the compound [10], the left-to-right conditioning of probabilities in n-grams is a bad fit to the underlying principle.

Although our approach also employs decompounding, it is important to note that it is substantially different from the large number of algorithms performing lexicon reduction. Instead we use the decompounding information to introduce new knowledge into the LM in order to model compounds when no data is available. As such, we intend to extend the vocabulary with new, unseen words and overcome the language modeling issues mentioned above.

3.2. Class-based n-gram models

The proposed technique is inspired by class-based n-gram models, as introduced by Brown et al [8]. 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:

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

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 however 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 n-gram model, where only unseen compounds are clustered together with their heads. In the next section we will argue that this clustering suffers less from the above issues.

4. SEMANTIC HEAD MAPPING

The issues introduced in Section 3.2 are less problematic for compounds, since they are well represented by their head, both syntactically and semantically. For most compound words, the head has the unique property of carrying inherent class information. This is obviously the case for the predominant class of endocentric compounds which introduce a hyponym-hypernym relation. It can be argued though that this is also true for copulative and appositional compounds. While these two types of compounds do not restrict the meaning of the compound, their heads can still be viewed as classes. The only troublesome compounds are exocentric compounds. However, because of their opaque meaning, they are in fact quite rare.

By mapping a compound onto its semantic head we effectively apply a clustering that does not depend on external information and can hence be applied to all compounds, regardless of their frequency in a training corpus. By clustering only the infrequent compounds, the obtained class-based n-gram reduces the risk of overgeneralization observed in most class-based LM approaches. This simplifies introducing new words and opens up possibilities for domain adaptation. To our knowledge this approach has not been described in

the literature for any language and is substantially different from the mentioned decompound-recompound approaches [11, 14, 15, 16] that fail to take advantage of the valuable semantic information embedded in compounds.

To obtain semantic heads for compounds, one could make use of existing morphological information. This information however proved to be insufficient for our needs, mostly because a semantic head can consist of more than one constituent. In addition, no morphological information is available for infrequent compounds, which are the main target of our technique.

In the following sections we therefore propose a fully automatic head mapper consisting of 2 parts: (1) a generation module which generates all possible decompounding hypotheses; and (2) a selection module which selects the most plausible head.

4.1. Generation module

First, all possible decompounding hypotheses are generated by means of a brute-force lexicon lookup: for all possible substrings w_1 and w_2 of the candidate compound w, $w = w_1 + w_2$ is an acceptable hypothesis if w_1 and w_2 are both in the lexicon. The substrings are optionally separated by the Dutch binding morphemes 's' and '-'. The module also works recursively on the first substring i.e. if w_1 is not in the lexicon, the module will verify whether or not it's a compound by itself. In its current implementation the system always makes the assumption that the head is located at the right-hand side of the compound, since this is almost exclusively the case for Dutch, as we discussed in Section 2.2. Hence, we do not expect this assumption to significantly influence the results.

We hypothesize that there is a significant discrepancy between the frequency of compound modifiers and heads: since a (endocentric) compound is typically a hyponym of its head and most if not all hypernyms have multiple hyponyms, the heads tend to occur frequently. Modifiers on the other hand are less frequent, because they constrain the hypernym to a more specific and often completely new domain e.g. *schaak+stuk (chess piece)*. To account for this discrepancy we allow the generation module to read from 2 different lexica: a modifier lexicon V_m and a head lexicon V_h. Although the 2 lexica can be filtered in any way, the current implementation only adopts word frequency filters. An exception is made for acronym modifiers consisting of all uppercase characters, which are automatically considered as valid words and are therefore not required to be lexical.

We further expect the amount of (false) hypotheses to increase drastically with decreasing constituent length which is especially true if the lexica contain (noisy) infrequent short words. Two parameters L_m and L_h are introduced to control the minimal length of modifiers and heads respectively.

4.2. Selection module

The generation module hugely overgenerates because it only has access to lexical knowledge. In the selection module we introduce knowledge based on corpus statistics to select the most likely candidate. Concretely, the selection between the remaining hypotheses is based on unigram probabilities and constituent length. We expect longer and more frequent constituents to yield more accurate results and provide selection parameters w_{len}, w_u and w_{pu} to weigh the relative importance of the head length, head unigram probability and product of the constituent unigram probabilities. We also considered the use of part-of-speech (POS) knowledge, but did not achieve any improvements with it, most likely due to incorrect POS tagging of the infrequent compounds.

Algorithm 1 shows pseudocode for the complete SHM algorithm, excluding the constituent separation by binding morphemes for the sake of clarity.

function GENERATE($compound, V_m, V_h, L_m, L_h$)
  for all $mod + head = compound$ **do**
    if $len(mod) \geq L_m$ **and** $len(head) \geq L_h$ **then**
      if $head \in V_h$ **then**
        if $mod \in V_m$ **or** $mod \in acronyms$ **then**
          $hypotheses \leftarrow (mod, head)$
        else
          $hypotheses \leftarrow (\text{GENERATE}(mod, ...), head)$
  return $hypotheses$
function SELECT_BEST($hypotheses, w_{len}, w_u, w_{pu}$)
  for all $(mod, head) \in hypotheses$ **do**
    $score \leftarrow w_{len} * length(head) + w_u * P_{uni}(head)$
      $+w_{pu} * P_{uni}(mod) * P_{uni}(head)$
    if $score > max\_score$ **then**
      $max\_score \leftarrow score$
      $best \leftarrow (mod, head)$
  return $best$

Algorithm 1: Semantic head mapping algorithm

5. PROBABILITY ESTIMATES

The compound-head pairs produced by the SHM algorithm can be used to enrich a language model with probability estimates for new, unseen compounds. To this purpose, the semantic head and all of its retrieved compounds are viewed as members of a single class. For each word in this class, the n-gram probability can be estimated as the product of a class n-gram probability and a within-class word probability, as was shown in Equation 1.

Since we have argued that a compound is well represented by its semantic head, we use the n-gram probability of the head as the class n-gram probability for each member. The within-class probability can be estimated by assigning a frequency count $\hat{c}(u)$ to each of the unseen compounds u and normalizing by the count of all members of the class C_{head}, defined by the semantic head:

$$P(u|C_{head}) = \frac{\hat{c}(u)}{c(head) + \sum_{u' \in C_{head}} \hat{c}(u')} \quad (2)$$

A sensible value for $\hat{c}(u)$ can be obtained empirically or more analytically, by averaging over the counts of all cut-off out-of-vocabulary (OOV) compounds with the same head i.e. the least frequent compounds with the same head which are cut off or disregarded during LM training. An alternative approach consists of distributing the probability mass uniformly within each class.

6. EXPERIMENTAL VALIDATION

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 [17] and integrated, together with the created LMs, into the recognizer described in [18]. The development data for the head mapper originates from CELEX [19] where the ground truth is based on a morphological analysis of 122k

types of which 68k are compounds. For each compound only one possible head is allowed which is optimal for most compounds, but might be too strict for others e.g. *borst+kanker+patiënt (breast cancer patient)* should be mapped to the semantically most similar head *kankerpatiënt (cancer patient)*, but a mapping to *patiënt (patient)* is still acceptable. The test data consists of the Flemish part of the Corpus Spoken Dutch [20] component o, which contains read speech. In order to focus on the efficiency of our proposed technique, the component was reduced to those fragments that contain OOV compounds for which a semantic head was retrieved. After reduction, the test data, which we will further refer to as CGN-o, contains almost 22h of speech. It consists of 192,153 tokens, produced by 25,744 types of which 1,625 are unseen in the LM training data and 953 are compounds.

6.1. Semantic head mapping

We applied an extensive grid search on the CELEX development data for all of the system parameters and counted the amount of true and false positives and negatives. We then calculated the precision and recall for each parameter setting and found that the optimal results were achieved with V_m=600k, V_h=200k, L_m=3, L_h=4, w_{len}=1, w_u=0 and w_{pu}=0. Table 1 shows that these parameters yield a precision of 80.31% and recall of 82.01% on the development data. When tested on the evaluation set, the precision is roughly equal with 80.25%, but the recall is even better with 85.97%. Moreover, many of the mappings that do not correspond to the ground truth are similar to the *borstkankerpatiënt* example. Although these mappings are suboptimal, they are nonetheless adequate, hence likely to have a positive impact on a LM.

6.2. LM integration

We trained initial, open vocabulary n-gram LMs of orders 2 to 5 with modified Kneser-Ney backoff on the 400k most frequent words in Mediargus. The remaining, cut-off OOV words were used to gather statistics for unseen words in a general OOV class. We then extended the 400k vocabulary with the unseen compounds for which the semantic head mapper found a valid head. This new, extended vocabulary was used when comparing WERs for the different estimation techniques.

As a baseline we considered two techniques that do not have the semantic head information at their disposal. Hence, these techniques have to resort to general OOV statistics i.e. the probability mass for the OOV class is redistributed over the newly added compounds using Equation 2, where all compounds are mapped to the OOV class instead of to their semantic head. The redistribution was done in two ways: uniformly and, analogous to section 5, based on the average cut-off OOV unigram count of all the compounds with the same head.

OOV-based mapping was compared to both the unigram-based and uniform SHM approaches, discussed in section 5. Although we also attempted to optimize $\hat{c}(u)$ empirically for both OOV-based and SH-based mapping, these results are not reported, as they did not invariably improve the results for all n-gram orders.

Table 2 shows the WERs of all these approaches, compared to the WERs of the initial LMs with 400k words, where no mapping was done. As can be seen, OOV-based mappings perform surprisingly well wrt the initial LMs which seems to indicate that lexicon extension is sufficient to recognize most of the unseen compounds. We suspect that this is due to the nature of our test set, which contains clean, read speech, and we expect this effect to be smaller with a more challenging data set. Unexpectedly, unigram-based OOV mapping also performs better than unigram-based SHM. Upon further investigation, we found that this was not caused by a low SHM n-gram coverage, but by an underestimation of $\hat{c}(u)$ due to the low counts of the cut-off OOV compounds, compared to the count of their heads. This shows that the unigram-based estimator is not reliable, as it is too dependent on the otherwise unused cut-off LM training data. The results for uniform SHM confirm this conclusion, as they produce a significant (Sign and Wilcoxon test $p < 0.0001$), relative WER reduction of approximately 1% over OOV-based mapping. This performance is more or less constant over the different n-gram orders and also shows in the perplexities where the relative improvement is about 6%.

CELEX (dev)		CGN-o (eval)	
precision	**recall**	**precision**	**recall**
80.31%	82.01%	80.25%	85.97%

Table 1. Semantic head mapping results as measured by precision and recall on CELEX and CGN-o

	n-gram order			
mapping technique	**2**	**3**	**4**	**5**
no mapping	31.31%	28.23%	27.59%	27.53%
uniform OOV	30.70%	27.67%	27.02%	26.97%
unigram-based OOV	30.63%	27.59%	26.96%	26.90%
unigram-based SHM	30.69%	27.65%	27.00%	26.95%
uniform SHM	30.33%	27.29%	26.65%	26.62%

Table 2. WERs for the initial 400k LMs (no mapping) and the different mapping techniques, as calculated on CGN-o

7. CONCLUSIONS AND FUTURE WORK

We introduced a new clustering technique to cope with language data sparsity by mapping compound words onto their semantic heads. Results on Dutch read speech show that our technique is capable of correctly identifying compounds and their semantic heads with a precision of 80.25% and a recall of 85.97%. A class-based language model with compound-head clusters achieves a significant, relative reduction in both perplexity and WER, of 6% and 1% respectively. We believe that SHM can have an even bigger effect on more spontaneous and/or noisy speech, which will be the subject of future investigation.

The approach is still suboptimal in the sense that we throw away any information from the modifiers. In the future we plan to investigate how we can take advantage of the modifier semantics. Also, in its current implementation we did not spend too much effort on the decompounding module, as this was not the main focus of our work. Better decompounding, including more accurate POS information, could improve the results further.

It would be interesting to investigate whether our technique can be extended to handle languages with a different lexical morphology. Romance languages are typically left-headed, applying the prepositional scheme mentioned in Section 2.1. In these languages, head mapping could then improve the prediction of the words following the compound instead of the compound itself.

Finally, we also plan to examine to what extent our technique could be beneficial for cut-off OOV compounds.

8. REFERENCES

[1] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[2] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. of Uncertainty in Artificial Intelligence*, 1999, pp. 289–296.

[3] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[4] 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.

[5] I. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, pp. 237–264, 1953.

[6] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proc. ICASSP*, 1995, vol. I, pp. 181–184.

[7] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Tech. Report TR-10-98, Computer Science Group, Harvard U., Cambridge, MA, August 1998.

[8] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, pp. 467–479, 1992.

[9] N. Ide, "Introduction to the special issue on word sense disambiguation: The state of the art," *Computational Linguistics*, vol. 24, pp. 1–40, 1998.

[10] G. Booij, *The morphology of Dutch*, Oxford University Press Inc., 2002.

[11] M. Nußbaum-Thom, A. El-Desoky Mousa, R. Schlüter, and H. Ney, "Compound word recombination for German LVCSR," in *Proc. Interspeech*, Florence, Italy, 2011, pp. 1449–1452.

[12] J. Zhou, Q. Shi, and Y. 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, K. Bali, A. G. Ramakrishnan, and P. P. Talukdar, "Automatic generation of compound word lexicon for Hindi speech synthesis," in *Proc. LREC*, 2004.

[14] T. Laureys, V. Vandeghinste, and J. Duchateau, "A hybrid approach to compounds in LVCSR," in *Proc. ICSLP*, 2002, vol. I, pp. 697–700.

[15] R. Ordelman, A. van Hessen, and F. de Jong, "Compound decomposition in Dutch large vocabulary speech recognition," in *Proc. Eurospeech*, Geneva, Switzerland, 2003.

[16] B. Reveil and J. Martens, "Reducing speech recognition time and memory use by means of compound (de-) composition," in *Proc. ProRISC*, 2008, pp. 348–352.

[17] K. Demuynck, T. Laureys, and S. Gillis, "Automatic generation of phonetic transcriptions for large speech corpora," in *Proc. ICSLP*, 2002, vol. I, pp. 333–336.

[18] K. Demuynck, A. Puurula, D. Van Compernolle, and P. Wambacq, "The ESAT 2008 system for N-Best Dutch speech recognition benchmark," in *Proc. ASRU*, 2009, pp. 339–343.

[19] H. Baayen, R. Piepenbrock, and L. 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/.