Reducing speech recognition time and memory use by means of compound (de-)composition

Bert Réveil and Jean-Pierre Martens
DSSP, ELIS, Ghent University
Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium
Email: {bert.reveil,jean-pierre.martens}@elis.ugent.be
Tel: +32 9 264 33 97, Fax: +32 9 264 35 94

***Abstract*—This paper tackles the problem of Out Of Vocabulary words in Automatic Speech Transcription applications for a compound language (Dutch). A seemingly attractive way to reduce the amount of OOV words in compound languages is to extend the AST system with a compound (de-)composition module. However, thus far, successful implementations of this approach are rather scarce.**

We developed a novel data driven compound (de-)composition module and tested it in two different AST experiments. For equal lexicon sizes, we see that our compound processor lowers the OOV rate. Moreover we are able to transform that gain in OOV rate into a reduction of the Word Error Rate of the transcription system. Using our approach we built a system with an 84K lexicon that performs as accurately as a baseline system with a 168K lexicon, but our system is 5-6% faster and requires about 50% less storage for the lexical component, even though this component is encoded in an optimal way (prefix-suffix tree compression).

I. Introduction

One of the performance boundaries of Automatic Speech Transcription (AST) systems is imposed by their vocabulary size. Words that are not in the system's lexicon, commonly named Out Of Vocabulary words, cannot be recognized. It is estimated that every OOV word invokes 1.6 to 2.2 transcription errors on average [1]. Therefore it is important to pursue the lowest possible OOV rate (measured on a large text corpus) given a system's resources.

As more and more processing power and memory becomes available, there is a tendency to add words to the vocabulary until the OOV rate drops below a threshold percentage. However, in embedded applications (eg. mobile speech technology) this approach might not be an option.

In several languages - and in Dutch in particular - one of the main causes of lexical variety is the frequent appearance of compound words. In fact, as the Dutch spelling conventions allow multiple combinations of orthographic units to form an orthographic unit on their own, compounding is considered to be an exponential mechanism, with the belief that people create new compounds every day [2]. A popular way to get the OOV rate down is therefore to split compounds into their constituent parts and to enter those parts in the vocabulary. An important condition for this approach to be successful in an AST application is that the constituents emerging from the AST system can be composed to compounds again afterwards. This is not easy because many constituent pairs also exist as word pairs.

An earlier attempt to elaborate the compound (de-)composition approach for Dutch speech transcription used empirically developed (de-)composition rules [2]. Others adopted a data driven decomposition approach [3] but they did not solve the composition problem.

We developed a novel data driven procedure for constructing a compound splitter on the basis of a word frequency list (derived from a text corpus) and a commercially available grapheme-to-phoneme (g2p) converter. Simultaneously, we conceived a composition module that relies on statistics being gathered during the compound splitting process. We tested the proposed compound processor on Flemish Broadcast News transcription tasks.

The rest of this paper is divided into two large parts. In section II we present the recipe behind our compound processing module. In section III we discuss the experimental validation of the proposed technique. At the end we draw some conclusions and make suggestions for future work.

TABLE I
N-BEST Flemish language modeling training data (Mediargus).

Newspaper	Years	Size (M words)
De Morgen	1999-2004	135
De Standaard	1999-2004	118
De Tijd	1999-2004	98
Gazet Van Antwerpen	1999-2004	240
Het Belang Van Limburg	1999-2004	106
Het Laatste Nieuws	1999-2004	284
Het Nieuwsblad	1999-2004	322
Het Volk	2000-2004	133
TOTAL VL	1999-2004	1436

II. Compound processing technique

Our data driven compound processor requires a large text corpus and a g2p converter. As text corpus we used a collection of normalized newspaper texts from the Flemish Mediargus corpus (Table I). The g2p converter is the Nuance g2p converter embedded in the autonomata g2p toolset [4]. In order to obtain Flemish transcriptions, we operated the g2p in DUB mode.

The compound splitter works in two phases. First, possible heads and tails of compounds are identified. Then, possible

compounds and their best head-tail decompositions are determined.

A. Finding the head and tail set

A word frequency list was derived from the Mediargus text data. To get rid of miss-spellings, we retained only words that appeared at least 20 times.

All found words (420K in total) and their word frequencies were encoded in a prefix tree. Every node of the tree, except the root node, contains the following information:

- a link to the previous node (useful for back-tracking)
- the letter on the incoming branch (idem)
- the cohort frequency = the number of times the node is being visited
- the word frequency = the number of times the node is the end node of a word
- a list of all successor nodes and the corresponding branch letters

In this tree we look for so called *head nodes* which satisfy the following four conditions, in which N_t represents the envisaged lexicon size.

1) There exists a word ending in that node, called the node word.
2) The node word belongs to the set of words that have a fair chance of making it to the lexicon after elimination of the compounds which will be reconstructed from their head and tail parts. This means that node words have to be in the top $N_t(1+\alpha)$, where α must presumably be in the range of 0.25..0.50.
3) The node word has a minimum length of 4 characters. In order to capture compounds like *wets+overtredingen* ('law violations') or *Unizo-+bestuurslid* ('board member of Unizo'), we relaxed the first condition and allowed K+1 character non-words consisting of a K letter word followed by a binding 's' or a '-' as well (K$\geq$3).
4) The node is not an end node. This implies that there is at least one other word that begins with the node word.

Tail nodes were obtained in exactly the same way, except that the tree is now constructed by presenting the words in reverse order (last letter first), and that tails can only be words of 4 characters or more.

For a target lexicon size of 42K (with α set to 0.25) we found 42.5K heads and 18.5K tails, while for an envisaged lexicon size of 84K we counted 66K heads and 25K tails.

B. Compound list generation

A word is a compound if the following conditions are met:

- It does not belong to the βN_t most frequent words. This condition ensures that very frequent words, which are likely to be recognized correctly, are not split.
- It can be decomposed into a valid head-tail combination.

The latter condition is more than a simple possibility check. A possible head-tail decomposition is considered valid if the automatic phonetic transcription of the compound is equal to the concatenation of the automatic phonetic transcriptions of the head and the tail. Otherwise the recognizer would not be able to correctly recognize the two parts of the compound. For instance: the phonetic transcription of the verb *verslappen* ('to weaken') is /v@rslAp@n/ whereas the phonetic transcription of the decomposition *vers+lappen* ('fresh+rags') would become /vErs/+/lAp@n/ and therefore an invalid decomposition.

Because too many plausible decompositions were rejected by the g2p filter we relaxed the rules a bit:

1) If the transcription of the head ends on /@n/ and this /n/ is dropped in the transcription of the compound with that head, the compound is accepted as well. This is intended for words like *juniorenkampioenschap* ('junior championship'). The transcription of this word is /jynior@kAmpiunsxAp/ whereas the transcription of the head *junioren* ('juniors') and the tail *kampioenschap* ('championship') leads to the concatenation /jynior@nkAmpiunsxAp/.
2) If the head of the compound is a word + a binding 's', the g2p converter is also asked to transcribe the head without the 's' and to add a /s/ to that transcription. If this transcription fits, the decomposition is accepted too, and this transcription will appear in the lexicon for the head part. This rule deals with words like *investeringstoelagen* ('investment appropriations'). The transcription of *investeringstoelagen* is /InvEsterINstulaG@n/. The transcription of the head *investerings* and the tail *toelagen* leads to /InvEst@rINstulaG@n/. By transcribing *investering* and adding an /s/, this problem can be fixed.

In case more than one valid decomposition is encountered, the decomposition with the highest head and tail value product is selected. The head value is defined as the cohort frequency of the head minus its word frequency. In other words: a head value indicates how frequent the head is seen as the head of a longer word. A tail value is defined in a similar way.

All found compounds are stored in a compounds file containing lines of the following form:

word - best head - best tail - word frequency

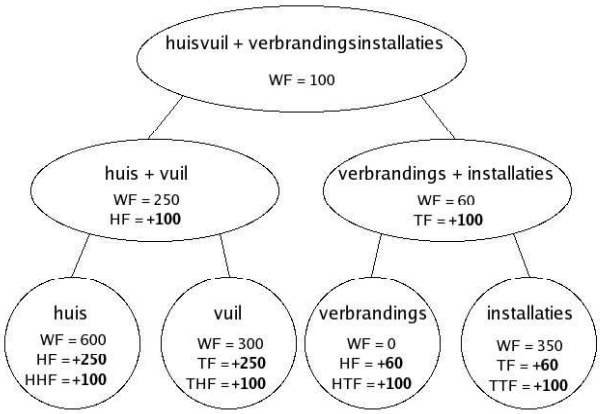
Following the above procedure and using β=0.25 we found 155K words of the 442K word list (35.1%) to be compounds for an envisaged lexicon size of 84K. For a 42K lexicon we ended up with 138.5K compounds (31.3%).

C. Extended frequency list

Once all possible compounds with their best decomposition are collected, a new frequency list must be determined. Thereby, it is assumed that all possible compounds consist of a maximum of 4 segments, meaning that a compound can be split into two parts and that these parts can be split a second time afterwards (see figure 1 for the word *huisvuilverbrandingsinstallaties* ('incinerators')). This assumption is loosely based on findings of [5].

Also in view of the composition step 7 frequencies were considered per word:

Fig. 1. Example of compound decomposition.



- word frequency (WF) = same as before
- head frequency (HF) = frequency with which the word appears as a head
- tail frequency (TF) = frequency with which the word appears as a tail
- head of head frequency (HHF) = frequency with which the word appears as head of a longer head
- tail of head frequency (THF) = frequency with which the word appears as tail of a longer head
- head of tail frequency (HTF) = frequency with which the word appears as head of a longer tail
- tail of tail frequency (TTF) = frequency with which the word appears as tail of a longer tail

One by one (the longest one first), the compounds were processed and the head and the tail lexicon trees were modified as follows (see also figure 1):

- In both trees the WF of the compound is set to 0 in the compound's end node.
- The corresponding HF and TF are increased with the compound's WF.
- The HHF of the head and the THF of the tail are increased with the HF of the compound (if > 0), while the HTF of the head and the TTF of the tail are increased with the TF of the compound (if > 0).

An extended frequency list was then extracted with entries of the following form:

lexicon item - WF - HF - TF - HHF - THF - HTF - TTF

The size of the 442K original Mediargus word list was reduced to 308K for a target lexicon size of 42K. For N_t=84K the word list size went from 442K to 293K.

D. Compound decomposition in text and vocabulary selection

By splitting all determined compounds appearing in the text corpus a new text version is generated. The target vocabulary is then selected by retaining the most frequent words in this text.

This text will also constitute the input for the language model development (see III).

E. Compound composition

Successive word pairs are merged to compounds based on the following two filters:

1) A correct composition must give rise to a compound appearing in the compounds list. Moreover, the composition must match the best decomposition recorded in that list.
   Obviously, this is a safe method but since people create new compounds every day, it may be too restrictive and therefore subject to future revisitation.
2) For each selected candidate compound, the frequencies in the extended frequency list are used to determine when to accept it or not. The decision is positive if:
   - either the head or the tail frequency (relative to the word frequency) exceeds a threshold γ, or
   - both the head and the tail frequency (relative to the word frequency) exceed a smaller threshold δ.

To make sure that compounds consisting of more than two parts can be formed, the compound composer runs twice over the recognized text.

TABLE II
DECOMPOSITION-COMPOSITION: RECALL, PRECISION AND F-MEASURE AS MEASURED ON MEDIARGUS TRAINING TEST SET.

N_t	Recall	Precision	F-measure
42K	97.39%	94.64%	96.00%
84K	95.92%	95.92%	95.92%

In order to fix γ and δ all compounds appearing in a small development set selected from the Mediargus text corpus were decomposed. Afterwards the composition module was applied to the decomposed text. By comparing the reference text with the output text, we were able to calculate recall and precision values. Recall was defined as the percentage of compounds that were correctly reconstructed. Precision was defined as the percentage of reconstructed compounds that were actually correct. For the list of compounds determined for a target lexicon size of 84K, we found that γ=25% and δ=1.5% gave a good recall and precision. For the 42K lexicon γ=50% and δ=1.0% gave the best results (Table II).

III. EXPERIMENTAL VALIDATION

The proposed compound processor was evaluated in the context of Flemish Broadcast News transcription. We performed tests on two different test sets. The first test set consists of 71 sentences uttered by 18 different speakers (approximately 1 hour of data) and was taken from the CGN corpus[1] that was recorded in the period 2000-2004, also covered by the Mediargus corpus. The second test set comprises 1482 utterances from 100 speakers (2 hours) and was taken from recent news broadcasts (after 2007).

[1] http://lands.let.kun.nl/cgn/home.htm

In all experiments the AST systems made use of cross-word triphone acoustic models trained on 53 hours of Flemish Broadcast News speech extracted from the CGN corpus. The AST systems were implemented with the HMM75 search engine [6].

For the sake of simplicity systems without compound processor are denoted as *BN01* systems in the following. Systems extended with the compound processing module are called *BN02* systems.

Open, 3-gram language models were constructed by means of the SRI LanguageModeling toolkit [7]. Based on Ordelman's findings [8] interpolated modified Kneser-Ney was chosen as the smoothing technique. The training material consisted of the normalized Mediargus newspapers texts listed in table I. BN02 language models were trained on the text versions with split compounds (see II-D).

A. Equal lexicon size

System vocabularies were determined by collecting the most frequent words appearing in the language model training data. For a lexicon size of 42K the BN01 training text coverage was 94.40%, while the BN02 coverage raised up to 96.20%. Table III shows the WER and the OOV rate on the two test sets obtained with the 42K recognizers. We can clearly see that both the OOV rate and the WER can be reduced significantly by applying the compound processor.

TABLE III
BN01 VS. BN02: TEST SET OOV RATES AND WER RESULTS FOR 42K LEXICONS.

Task	System	OOV rate	WER
BN-VL-dev	BN01	3.60%	21.4%
BN-VL-dev	BN02	2.02%	19.2%
BN-VL-eval	BN01	3.46%	29.8%
BN-VL-eval	BN02	2.54%	28.9%

A lexicon size of 84K led to a BN01 training text coverage of 96.61%, while for the BN02 texts this coverage increased to 97.92%. Table IV depicts the test set OOV rates and WERs for the 84K systems. Again, the results indicate that we are able to lower the WER.

Although one might conclude from the data that a 42K BN02 system performs as accurately as an 84K BN01 system, we want to stress that this conclusion may not be entirely legitimate as the language models used for the 42K systems were somewhat larger than the ones we used for the 84K systems. Experiments with language models of the same size are in progress.

TABLE IV
BN01 VS. BN02: TEST SET OOV RATES AND WER RESULTS FOR 84K LEXICONS.

Task	System	OOV rate	WER
BN-VL-dev	BN01	1.94%	19.2%
BN-VL-dev	BN02	1.06%	18.4%
BN-VL-eval	BN01	2.38%	28.8%
BN-VL-eval	BN02	1.73%	28.3%

B. Reduction of memory use and recognition time

In order to investigate whether our compound processor brings other advantages than a gain in recognition accuracy, we built a BN01 system with a 168K lexicon and compared it to the 84K BN02 system in terms of recognition accuracy, recognition time and memory use needed to store the system's lexicon. As we wanted to make a fair comparison, we chose to use the same bigram and trigram cut-off values for the language model, ensuring that the language models were of a comparable size. Table V shows the results.

TABLE V
BN01 - 168K VS. BN02 - 84K: TEST SET OOV RATES, WERS, RECOGNITION TIME AND MEMORY USE.

Task	System	OOV rate	WER	xRT	Mem. (B)
BN-VL-dev	BN01	0.99%	18.2%	2.61	14.656.403
BN-VL-dev	BN02	1.06%	18.4%	2.45	7.409.019
BN-VL-eval	BN01	1.70%	28.2%	4.11	14.656.403
BN-VL-eval	BN02	1.73%	28.3%	3.90	7.409.019

We conclude that a BN02 system with a 84K lexicon reaches the same accuracy as a BN01 system with a 168K lexicon, but the BN02 recognizer is 5-6% faster and requires 49.5% less memory to store the lexical component. The latter is remarkable as the HMM75 recognition engine already uses prefix-suffix tree compression to encode that component. Therefore we believe that our compound processor might be a useful tool in embedded applications where time and memory space are of a greater importance.

IV. CONCLUSIONS AND FUTURE WORK

In this work a novel data driven compound (de-)composition module was proposed. For a fixed lexicon size, the module causes a significant WER drop in an AST application. The compound processor also allows the AST system to achieve a certain WER at a much lower cost: 5-6% less CPU-time and 49.5% less memory for the lexical component.

As our approach is data driven we believe that it can be applied to other compound languages such as German or Finnish with only minor modifications.

In order to achieve better results we can consider to also allow non-words like *afge* in *afge+lopen* ('finished') or *elijk* in *hart+elijk* ('cordial') as a head or a tail. This might lead to an even bigger reduction of the OOV rate, and thus of the WER rate. In fact, as one knows that these non-words cannot exist as separate words, no extra composition errors should be expected.

Allowing unseen compounds to be formed is a second useful extension.

ACKNOWLEDGMENT

The presented work was carried out in the context of two research projects: the N-BEST project, granted under the Dutch-Flemish STEVIN program, and the TELEX project, granted by Flanders FWO.

REFERENCES

[1] M. Adda-Decker and L. Lamel, "The use of lexica in automatic speech recognition," in *Lexicon Development for Speech amd Language Processing*. Frank Van Eynde and Dafydd Gibbon, Eds. (Kluwer), 2000, pp. 235–266.

[2] T. Laureys, V. Vandeghinste, and J. Duchateau, "A hybrid approach to compounds in LVCSR," *Proceedings ICLSP (Denver)*, pp. 697–700, 2002.

[3] R. Ordelman, A. van Hessen, and F. de Jong, "Compound decomposition in Dutch large vocabulary speech recognition," in *Proceedings of Eurospeech 2003, Geneve*, 2003.

[4] Q. Yang, J. Martens, N. Konings, and H. van den Heuvel, "Development of a phoneme-to-phoneme (p2p) converter to improve the grapheme-to-phoneme (g2p) conversion of names," *LREC Conference*, 2006.

[5] C. Monz and M. de Rijke, "Shallow Morphological Analysis in Monolingual Information Retrieval for Dutch, German and Italian," *Lecture Notes In Computer Science; Vol. 2406*, pp. 262–277, 2001.

[6] K. Demuynck, J. Duchateau, D. V. Compernolle, and P. Wambacq, "An Efficient Search Space Representation for Large Vocabulary Continuous Speech Recognition," *Speech Communication; Vol. 30, No. 1*, pp. 37–53, 2000.

[7] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado*, September, 2002.

[8] R. Ordelman, *Dutch Speech Recognition in Multimedia Information Retrieval*. Taaluitgeverij Neslia Paniculata, Enschede, 2003.