

Handling Compound Nouns in a Swedish Speech-Understanding System

David Carter¹, Jaan Kaja², Leonardo Neumeyer³, Manny Rayner¹, Fuliang Weng³ and Mats Wirén²

¹SRI International
Suite 23, Millers Yard
Cambridge CB2 1RQ, UK

²Telia Research AB
Spoken Language Processing
S-136 80 Haninge, Sweden

³SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025, USA

ABSTRACT

This paper describes and evaluates a simple and general solution to the handling of compound nouns in Swedish and other languages in which compounds can be formed by concatenation of single words. The basic idea is to split compounds into their components and treat these components as recognition units equivalent to other words in the language model. By using a principled grammar-based language-processing architecture, it is then possible to accommodate input in split-compound format.¹

1. INTRODUCTION

In many languages, including German, Dutch, Swedish, Finnish and Greek, compound nouns can be formed by concatenation of single nominals. For example, in Swedish “folk” and “musik” can be put together to form “folkmusik” (*folk music*), and this in turn can be combined with “grupp” to form “folkmusikgrupp” (*folk music group*), and so on. In most previously reported work, such as the widely publicized SQALE project, compounds have been treated in the same way as any other words. However, as the vocabulary size grows, the productive nature of compounding makes this kind of approach increasingly less feasible; the most obvious indication of the problem’s seriousness is the magnitude of the out-of-vocabulary (OOV) rate. For example, in an experiment on SQALE training texts [8, page 186], using 20 000-word lexicons for both German and English resulted in a 7.5 % OOV rate for German and 2.5 % for English. The German lexicon had to be extended to 64 000 words to obtain OOV rates similar to those of the 20 000-word lexicon for English.

Results like those quoted above strongly suggest that treating compounds in the same way as other words is not satisfactory. Two recent papers address this issue. Spies [12] reports results on an isolated-word large-vocabulary German dictation application, in which the components of compounds, rather than the compounds themselves, were treated as units. Geutner [7] describes a more elaborate method, also implemented for German, in which full morphological decomposition of words was used. Both authors report unspectacular but encouraging initial results.

This paper describes and evaluates a simple and general solution to the handling of Swedish compound nouns, carried out in the spirit of the work reported in the two above-mentioned papers. Syntactically, semantically and phonologically, Swedish compound nouns are similar to English compound nominals except for the obvious difference: English orthography inserts spaces between components, while Swedish omits them. These observations suggested to us that Spies’ strategy should be an appropriate way to attack the problem: splitting compounds into their components, and treating these components as recognition units equivalent to other words in the language model. In contrast to Spies, however, our recognizer is for continuous speech, and is embedded in a spoken language understanding system. It is thus necessary not only to recognize, but also to reassemble and make sense of the split compounds, the means to do this being provided by the language-processing modules of the system.

We believe that our methods should handle verbal and other kinds of compounds equally well; however, since noun compounding in Swedish is more productive, and the other kinds of compound less common in our material, we have chosen here to concentrate on nouns. Inflectional morphology is a less serious problem in Swedish than in German (in particular, Swedish verbs are not inflected by either number or person). For this reason, we decided that full morphological decomposition *à la* Geutner would probably not justify the additional complexity introduced.

Our approach has been fully implemented within a Swedish-to-English version of the Spoken Language Translator [10, 11]. The system is capable of translating spoken utterances from the Air Travel Planning (ATIS) domain from Swedish into English, using a vocabulary of about 1 500 words. Continuous speaker-independent speech recognition is performed by a Swedish version of the DECI-PHER (TM) recognizer [9], and language processing is provided by a Swedish version of the SRI Core Language Engine [3, 6].

The rest of the paper is organized as follows: Section 2. describes the corpus material used for other experiments. Section 3. describes experiments involving the Swedish speech recognizer alone, and Section 4. describes further experiments on the full speech translation system. Section 5. presents our conclusions.

¹The work reported here was funded by Telia Research AB under the SLT-2 project.

2. CORPUS

The current version of the SLT system operates in the ATIS domain. For English, there is a carefully collected corpus of about 20 000 utterances. No corresponding corpus existed for Swedish when the current project started in 1995. We have gone through several iterations of creating successively more realistic Swedish versions of the ATIS corpus. The experiments described here were performed using Version 1 of Swedish ATIS (hereafter “ATIS-S-1”). A second version of Swedish ATIS, constructed since then, is described briefly in Section 5., and a third version is currently being collected.

ATIS-S-1 was produced by the following process. First, a set of about 5 000 original English ATIS utterances was randomly selected from the full English ATIS corpus. Four randomly selected subsets, each of about 2 800 sentences, were then each translated into Swedish by each of four different Telia employees. Finally, the resulting Swedish sentences were divided, roughly equally, between 100 native speakers of the Stockholm dialect of Swedish for reading and recording. In total, 11 275 sentences were recorded, of which 10 831 sentences were used for training and 444 held out for testing. The primary goal of this initial corpus collection effort was rapid creation of training material for a first version of the Swedish recognizer. A secondary goal was to provide basic text resources for use in the development of the Swedish language-processing modules.

The orthographic transcriptions of the ATIS-S-1 sentences were further processed to create two different versions of the text corpus. In the first, “split” version, all compound words, including numbers, were split into their components. In the second, “unsplit” version, only numbers were split. In both cases, the numbers were split because it would be futile to try to list them in the lexicon; the same approach was taken in the German SQALE experiments [8, page 186].

The split and unsplit versions of the ATIS-S-1 text were used to train two different versions of the Swedish recognizer. The two versions of the recognizer differed only in terms of vocabulary and language model. The recognition vocabulary consisted of the set of all surface words in the relevant version of the corpus. The bigram language model was calculated directly from the corpus, without, for example, backing off surface words to classes.

3. SPEECH-RECOGNITION EXPERIMENTS

To compare the split and unsplit approaches at the recognition level, we performed two experiments with respect to word-error rate (WER), using data from the full set of 444 test sentences with 3 584 unsplit words and 3 758 split words. In one experiment, split training data and a split lexicon were used for language modeling; in the other, unsplit training data and an unsplit lexicon were used. The results are shown in Table 1.

Since the total number of words is different in the split and unsplit cases, the WER with respect to compounds can be measured in two ways. More specifically, the following two methods for calculat-

	Split	Unsplit
WER with respect to compound components (method 1)	7.9 %	8.2 %
WER with respect to full compounds (method 2)	8.3 %	8.7 %

Table 1: Word-error rates obtained in the experiments.

ing the WER were used: In the first one, corresponding to the first row of Table 1, a splitting function, which (for the purpose of the experiments) is used for mapping compounds to their components, was applied to both the hypotheses from the recognizer and to the references. (This function modifies only the unsplit data.) We then compared the newly formed hypotheses and references to get the WER. Thus, in this case the WER was calculated with respect to the compound components.

In the second method, corresponding to the second row of the table, the same splitting function, but with mappings of numbers removed, was used in the reverse direction to map all the compound components in both hypotheses and references back to their compounds. The result was then used for computing the WER. Thus, in this case the WER was calculated with respect to the full compounds.

From the point of view of language processing, it is the main (bold-faced) diagonal that is of primary relevance, since what we want to compare are recognizers that output either split or unsplit words. These figures show a modest improvement in WER. The other diagonal has been included to provide a fair comparison from the point of view of speech-recognition performance.

As for the unsplit case in method 1, we have the methodological problem that compound words as well as their components are in the recognizer lexicon. There is thus a good chance that the recognizer will output the components, whereas the reference contains the compound. This will then be counted as one substitution followed by one or more insertions. It can be argued that the confusion between a compound and its components is not a major error. Method 1, applied to the unsplit case, removes this ambiguity and gives a performance figure that can be compared with the split case.

Method 2 can be said to simulate a language-processing system in the sense of reassembling the compounds. In most cases, this reverse mapping is straightforward, but there are cases in which a potential compound does not actually constitute a compound where it occurs in a sentence. For example, the temporal noun phrase “måndag eftermiddag” (*Monday afternoon*) has a corresponding compound “måndageftermiddag”. However, the two forms differ in meaning (and are also prosodically distinct). Since method 2 creates a compound from every sequence of words that in some context could be a compound, it does not take this difference into account, and in this sense the figures in the second row of Table 1 are imperfect.

As a comparison, the WER in the unsplit case *without* the reverse

mapping is 9.3%. This number is relevant to tasks like dictation, where a confusion between a compound and its components would be considered an error.

4. SPLIT VS. UNSPLIT COMPOUNDS IN SPEECH UNDERSTANDING

The results in Section 3 show that compound splitting produced a modest improvement in the recognizer’s WER. In the context of a speech-understanding system like SLT, however, the most relevant criterion for success is the effect on end-to-end performance. Another question of practical importance is the extent to which the language-processing modules need to be altered to accommodate input in split-compound form.

4.1. End-to-end performance evaluation

To test the effect of compound splitting on end-to-end system performance, we used the 444-sentence test set as input to two experiments involving language processing as well as speech recognition. The two sets of N-best speech hypothesis lists were each processed through the successive stages of Swedish language analysis, Swedish-to-English transfer, and English language generation. Finally, the two sets of English outputs were pairwise compared. Language processing was carried out using a robust fallback mechanism (described elsewhere), so that a translation was always produced.

We have noticed when testing and demonstrating the SLT system that people give widely different judgments as to whether a translation is “acceptable”. Indeed, it seems unlikely to us that this notion can be given a clear definition independent of a specific context of use. We have also observed, however, that there is much greater agreement on the *relative* quality of different translations. Given two candidate translations of the same utterance, it is normally not controversial to claim that one is better than the other, or that they are in practice equally good.

Our experiments involved 444 test utterances, 41 of which gave rise to different translations when compound splitting was introduced. These 41 utterances were examined by three independent judges, who were all native speakers of English and fluent in Swedish. Each utterance was presented together with the two candidate translations produced by the “split” and “unsplit” versions of the system, respectively: each judge was asked to state whether translation 1 was better or worse than translation 2, or alternatively that neither translation was clearly better than the other. The order in which the two translations were presented—that is, “split” before “unsplit” or *vice versa*—was decided randomly in each case. The results are summarized in Table 2. Agreement between the judges was good: in only five sentences out of the 41, a pair of judges gave opposite judgments, one marking the split version as better and the other marking it as worse.

It is interesting to note that although the unsplit version was in several cases better than the split one, the errors in the split translations were never actually caused by failure on the part of the CLE (see Section 4.2) to reassemble a split compound.

	Split better	Unsplit better	Unclear
Judge 1	19	12	10
Judge 2	24	11	6
Judge 3	19	10	12

Table 2: End-to-end evaluation comparison, giving each judge’s preferences for utterances where the translation was affected by compound splitting.

4.2. Language processing for split compounds

Language processing in SLT is carried out by the Core Language Engine (CLE), a general language-processing system, which has been developed by SRI International in a series of projects starting in 1986. The original system was for English only. The Swedish version [6] was developed in a collaboration with the Swedish Institute of Computer Science. The CLE is extensively described elsewhere [1, 2, 3], so we only give the minimum background necessary for understanding our handling of compounds.

The basic functionality offered by the CLE is two-way translation between surface form and a representation in terms of a logic-based formalism called Quasi Logical Form (QLF). The modules constituting a version of the CLE for a given language can be divided into three groups, which we refer to as “code”, “rules,” and “preferences”. The “code” modules constitute the language-independent compilers and interpreters that make up the basic processing engine; the other two types of module between them constitute a declarative description of the language.

The “rules” contain domain-independent lexico-grammatical information for the language in question; they encode a relationship between surface strings and QLF representations. Thus, for any given surface string, the rules define a set of possible QLF representations of that string. Conversely, given a well-formed QLF representation, the rules can be used to produce a set of possible surface-form realizations of the QLF. The code modules support compilation of the rules into forms that allow fast processing in both directions: surface-form \rightarrow QLF (analysis) and QLF \rightarrow surface-form (generation).

The relationship between surface form and QLF is in general many-to-many. “Preference” modules contain data in the form of statistically learned distributional facts, based on analysis of domain corpora [4]. Using this extra information, the system can distinguish between plausible and implausible applications of the rules with a fairly high degree of accuracy.

The principled grammar-based architecture of the CLE made it simple to modify the speech-language interface [5] to accommodate input in split-compound format. Since morphology and syntax rules have the same form [3, §3.9], all that was necessary was to change the status of compounding rules from “morphology” to “syntax”. In a little more detail:

- Declarations were supplied to identify some morphology rules

as specifically compounding rules.

- A switch was added, which, when On, allowed the designated morphology rules to be used as syntax rules.

After a little experimentation, it also turned out to be advantageous to add a few dozen lexicon entries, to cover words that could potentially be constructed as compounds, but in reality are noncompounds. These were automatically generated from the lexicon by using a simple algorithm. No other changes to the system were made, and the adaptation process required only two person-days of work.

5. CONCLUSIONS

To summarize the results of our experiments on ATIS-S-1, we found that compound splitting introduced an undramatic but tangible improvement in both WER and end-to-end system performance. It decreased the vocabulary size for both speech and language processing, and required no substantial modification of any part of the system. Our overall conclusion is that it is a clear win.

We were nonetheless somewhat disappointed to find that the improvement resulting from compound splitting was not larger. We believe that one reason for this lack of improvement was the very small number of translators used to create ATIS-S-1, which led to an unnaturally uniform and homogeneous corpus; in particular, the OOV rate on the test portion, even without compound splitting, is only about 0.5%. Preliminary results on a new version of Swedish ATIS, ATIS-S-2, support this hypothesis.

ATIS-S-2 has been created in roughly the same way as ATIS-S-1, but using a much larger number of translators, 427 in all. The result, a text corpus containing 4 592 sentences, is a considerably more reasonable approximation to a “real” Swedish ATIS corpus. We merged the ATIS-S-1 and ATIS-S-2 corpora, taking half of ATIS-S-2 as test data and the remaining material as training. Examining this new data, about 5% of all tokens in the test set are compounds, and the OOV rate of the full test set is 3.0%. In contrast, the OOV rate measured just on compounds is nearly 23%. However, if compounds are split, OOV falls from 3.0% to 2.1% (30% relative) on the whole test-set, and from 23% to 7% (70% relative) on compounds only. The above statistics give us reason to expect that the effect of compound-splitting on WER and end-to-end performance would be rather greater on a more realistic corpus. We hope to have tested this conjecture properly by the time of the conference.

6. REFERENCES

1. M.-S. Agnäs, H. Alshawi, I. Bretan, D.M. Carter, K. Ceder, M. Collins, R. Crouch, V. Digalakis, B. Ekholm, B. Gambäck, J. Kaja, J. Karlgren, B. Lyberg, P. Price, S. Pulman, M. Rayner, C. Samuelsson, and T. Svensson. Spoken Language Translator: First Year Report. SRI technical report CRC-043 (also SICS research report R94:03), SRI International, Cambridge, England, 1994. Available through WWW from <http://www.cam.sri.com>.
2. H. Alshawi, D. Carter, R. Crouch, S. Pulman, M. Rayner, and A. Smith. CLARE: A Contextual Reasoning and Cooperative Response Framework for the Core Language Engine. SRI technical report CRC-028, SRI International, Cambridge, England, 1992. Available through WWW from <http://www.cam.sri.com>.
3. Hiyani Alshawi, editor. *The Core Language Engine*. MIT Press, Cambridge, Massachusetts, 1992.
4. Hiyani Alshawi and David Carter. Training and Scaling Preference Functions for Disambiguation. *Computational Linguistics*, 20(4), 1994.
5. D. Carter and M. Rayner. The Speech-Language Interface in the Spoken Language Translator. In *Proc. 8th Twente Workshop on Language Technology*, University of Twente, Enschede, the Netherlands, 1994.
6. B. Gambäck and M. Rayner. The Swedish Core Language Engine. In *Proc. Third Nordic Conference on Text Comprehension in Man and Machine*, Linköping, Sweden, 1992. Also SRI Technical Report CRC-025. Available through WWW from <http://www.cam.sri.com>.
7. P. Geutner. Using Morphology Towards Better Large-Vocabulary Speech Recognition Systems. In *Proc. ICASSP 95*, 1995.
8. L. Lamel, M. Adda-Decker, and J. L. Gauvain. Issues in Large Vocabulary, Multilingual Speech Recognition. In *Proc. Eurospeech '95*, pages 185–188, 1995.
9. H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub. Large Vocabulary Dictation using SRI's DECIPHER(TM) Speech Recognition System: Progressive Search Techniques. In *Proc. ICASSP 93*, 1993.
10. M. Rayner, H. Alshawi, I. Bretan, D.M. Carter, V. Digalakis, B. Gambäck, J. Kaja, J. Karlgren, B. Lyberg, P. Price, S. Pulman, and C. Samuelsson. A Speech to Speech Translation System Built From Standard Components. In *Proc. 1st ARPA Workshop on Human Language Technology*, 1993.
11. M. Rayner, I. Bretan, D. Carter, M. Collins, V. Digalakis, B. Gambäck, J. Kaja, J. Karlgren, B. Lyberg, P. Price, S. Pulman, and C. Samuelsson. Spoken Language Translation with Mid-90's Technology: A Case Study. In *Proc. Eurospeech '93*, 1993.
12. Marcus Spies. A Language Model for Compound Words in Speech Recognition. In *Proc. Eurospeech '95*, pages 1767–1770, 1995.