Unsupervised and Knowledge-Free Learning of Compound Splits and Periphrases

Florian Holz and Chris Biemann

NLP Group, Department of Computer Science, University of Leipzig
{holz|biem}@informatik.uni-leipzig.de

Abstract. We present an approach for knowledge-free and unsupervised recognition of compound nouns for languages that use one-word-compounds such as Germanic and Scandinavian languages. Our approach works by creating a candidate list of compound splits based on the word list of a large corpus. Then, we filter this list using the following criteria:
(a) frequencies of compounds and parts,
(b) length of parts.

In a second step, we search the corpus for periphrases, that is a reformulation of the (single-word) compound using the parts and very high frequency words (which are usually prepositions or determiners). This step excludes spurious candidate splits at cost of recall. To increase recall again, we train a trie-based classifier that also allows splitting multi-part-compounds iteratively.

We evaluate our method for both steps and with various parameter settings for German against a manually created gold standard, showing promising results above 80% precision for the splits and about half of the compounds periphrased correctly. Our method is language independent to a large extent, since we use neither knowledge about the language nor other language-dependent preprocessing tools.

For compounding languages, this method can drastically alleviate the lexicon acquisition bottleneck, since even rare or yet unseen compounds can now be periphrased: the analysis then only needs to have the parts described in the lexicon, not the compound itself.

1 Introduction

A number of languages extensively use compounding as an instrument of combining several word stems into one (long) token, e.g. Germanic languages, Korean, Greek and Finnish. Compared to languages such as English, where (noun) compounds are expressed using several tokens, this leads to a tremendous increase in vocabulary size. In applications, this results in sparse data, challenging a number of NLP applications. For IR experiments with German, Braschler et al. report that decompounding results in higher text retrieval improvements than stemming [1].

As an example, consider the German compound "Prüfungsvorbereitungsstress" (stress occurring when preparing for an exam) - without an analysis, this word can neither be translated in an MT system nor found by a search query like "Stress AND Prüfung" (stress AND exam).

A. Gelbukh (Ed.): CICLing 2008, LNCS 4919, pp. 117–127, 2008.

Several approaches have been used to alleviate this threat by analyzing and splitting compounds into their parts, which will be reviewed in the next section. Then, we describe our approach of finding not only the correct splits, but also periphrases (a sequence of tokens with the same semantic interpretation, usually a noun phrase). Especially for long compounds, these periphrases exist in large corpora - our example is more commonly expressed as "Stress bei der Prüfungsvorbereitung".

During the whole process, which his described in Sect. 2, we do neither assume the existence of language-specific knowledge nor language-specific preprocessing tools. We argue that before augmenting the process with additional sources of information that might blur evaluation of the basic method, we first want to provide a language-independent baseline.

Section 3 presents experimental results for German, Section 4 concludes.

1.1 Literature Review

Approaches to compound noun splitting can be divided into knowledge-intensive and knowledge-free approaches. In knowledge-intensive approaches, either supervised learning from a training set is used to create a splitter, or a set of handcrafted rules performs the task. Not surprisingly, most studies on compound noun splitting report experiments with German, which is the compounding language with most speakers. For German compounding rules, see [6].

Knowledge-free approaches are in principle independent of language-specific knowledge and try to induce a compound splitter by analyzing the word list (types) of a raw text corpus.

Perhaps the most straightforward knowledge-free approach is described in [8]: In the analysis of a compound candidate, all prefixes of the candidate are matched against the word list. Once a prefix is found, a split is performed and the remaining suffix is subject to further analysis. The authors report 60% precision and 50% recall evaluation on correct split positions for a set of around 700 complex nouns. The main problem of this approach is caused by too many splits due to short words in the word list (e.g. "Prüf" for the example) that also cause the subsequent splits to fail. What would be needed to repair spurious splits are more clues about the semantic composition of compounds. Nevertheless, significant improvements in a retrieval task were obtained.

Larson et al. train a letter-n-gram classifier on word boundaries in a large corpus and use it to insert compound part boundaries, successfully reducing the out-of-vocabulary rate in their speech recognition system, but not improving speech recognition accuracy [7]. Here, no evaluation on the compound splitting itself is given.

Our approach falls into the knowledge-free paradigm. In the remainder of this section, we discuss some knowledge-intensive methods, like handcrafted or trained morphological analyzers, for completeness.

E. g. Finkler et al. provide several splitting options and leave the choice to a post-processing component [4]. Comparable to approaches for the related task of Chinese word segmentation, Schiller uses weighed finite-state transducers,

resulting in over 90% precision and almost perfect recall [10]. Sjöbergh et al. modified a spell checker to find and split compounds in a small Swedish text with very good results [11]. Yun et al. pursue a hybrid rule-based and statistical approach for Korean Compound noun splitting [14].

When translating between compounding and non-compounding languages in a Machine Translation system, productive compounds cannot be enumerated in the word list; moreover, they usually translate to two or more words or even phrases [5]. Parallel text can be employed to get clues about translations of compounds and to use the back-translation of parts for splitting, as in [2] and [5], who report over 99% accuracy when using POS-tagged parallel corpora. However, these methods need aligned parallel text, which might not be sufficiently available for all compounding languages in all relevant domains.

We know about one unsupervised approach to judge the suitability of descriptive patterns for given word pairs [12]. Here corpus-based frequencies of the word pairs and the patterns in which the word pairs appear in the corpus are used to rank the found or given patterns for a word pair. The patterns are taken to represent the relation between the to words. So for a word pair its inner relation can be identified using best fitting pattern or for a word pair another word pair out of a given set can be identified to resemble the same relation (SAT test). The evaluation of the first task is done with 600 manually labelled word pairs where for every pair the other 599 serve as training data to classify the choosen one. The other task is evaluated on 374 college-level multiple-choice word analogies. Both evaluations show results between 50% and 60% for precision, recall and the F1-measure.

1.2 Motivation for Our Method

Even if correct splits can be found automatically with high precision, the question arises how to interpret the results. Dependent on the application, extra information might be needed. Consider e.g. the two German compounds "Schweineschnitzel" (pork cutlet) and "Kinder-schnitzel" (small cutlet for kids, literally kids cutlet). In e.g. semantic parsing, it might be advantageous to know that a "Kinderschnitzel" is made for kids and not out of kids, as in the case of pork. We therefore propose to find periphrases, e.g. "Schnitzel vom Schwein" and "Schnitzel für Kinder" and offer these to the interpreting engine. Periphrases do not only add more valuable information, they will also be employed to find the correct splits: If no periphrasis for a candidate split exists, it is more likely to be a spurious candidate. To the best of our knowledge, no previous approaches to automatically finding periphrases without using manual resources for compounds are reported in the literature.

2 Method

In this section we outline the steps of our method, which builds entirely on a word list with frequencies and a sentence list obtained from a massive raw text corpus. Parameters in the process are explained in the moment they arise first in our outline and summarized in Tab. 1. The whole workflow is shown in Fig. 1.

Table 1. Resources, parameters and abbreviations

Resources	corpus word list	CWL
	corpus word frequencies	CWF
	corpus sentence list	CSL
Parameters	minimum morpheme length	mM
	minimum word (compound) frequency	mFr
	minimum morpheme frequency	mTFr
	number of split parts	mA
	distance between parts in periphrases	d

Preprocessing. As preprocessing, we assume a word list (types) with frequencies from a massive monolingual raw corpus in lowered capitalization. Since we do not use parts-of-speech or other information, we always operate on the full list, where words with dashes and other special characters, as well as numbers, are removed beforehand.

Candidate Split Extraction. As a first step, we try to find candidate splits generating all possible splits and checking whether the resulting parts are contained in the word list. Candidate split determination is parameterized by a minimum length of parts (mM) and by a minimum frequency for the compound candidate (mFr), arguing that very rare words are more likely to be typing errors than valid compounds. This is compareable to the approach undertaken by [8].

Candidate Filtering. The list of candidates is then filtered according to various criteria. We applied a maximum number of parts (mA), a minimum frequency of parts (mTFr). If several splits for one compound candidate exist, we select the split with the highest geometric mean of part frequencies (quality measure taken from [5]).

After this step, we have a list of compounds with splits that can already be evaluated. Results characteristically show high precision but low recall – The filtering succeeds in finding good splits, but only for a small fraction of total types (cf. Tab. 6 and 7).

Generalized Splitter. To overcome the recall problem, we train two trie-based classifiers on the filtered candidate splits to generalize possible parts over the full word list. If we, for example, included the correct split "Kinder-schnitzel" but not "Schweine-schnitzel", the second split will be classified based on the assumption that splitting the suffix "schnitzel" is generally correct. Training is done for prefixes and suffixes separately and allows constructing a recursive splitter that can be applied to arbitrary word lists. Table 2 shows a small training sample for the prefix and suffix classifier. When classifying, the input is matched against the longest prefix (suffix) stored in the trie, and the corresponding class distribution is returned. Classes denote here the number of letters that should be separated from the beginning (ending). Note that the suffix classifier internally works on reversed strings. For more detailes on patricia-tree-based classifying, see [13].

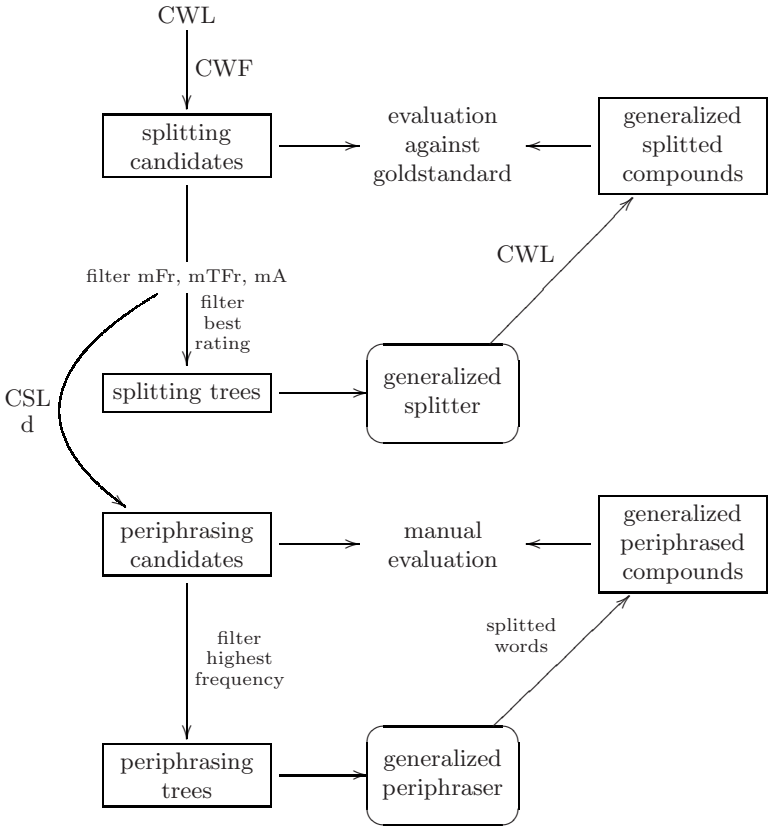

Fig. 1. Workflow: From the corpus word list (CWL), we build candidate splits and a generalized splitter; both are evaluated against the gold standard splits. From the candidate splits, we obtain periphrasis candidates and build a generalized periphraser; both are evaluated manually.

Table 2. Training set example for the prefix and suffix trie-based classifiers

Word	Split	Prefix String	Prefix Class	Suffix String	Suffix Class
Holzhaus	Holz-haus	Holzhaus	4	suahzloH	4
Berggipfel	Berg-gipfel	Berggipfel	4	lefpiggreB	6
Hintergedanke	Hinter-gedanke	Hintergedanke	6	eknadegretniH	7

Periphrase Detection. For a list of candidate splits, be it the list after candidate filtering or the list resulting from the application of the generalized splitter on the full word list, we set out to find periphrases in our corpus. Periphrases contain all parts, with at most d function words between them as intermediate fillers. For approximating function words, we allow the 200 most frequent words in this position.

For each candidate periphrasis per compound, we accept the periphrasis with the highest corpus frequency. Naturally, not all compounds are actually expressed as periphrases in the corpus. If strong evidence for a periphrasis is found, then the corresponding candidate split can be assumed to be valid with even higher confidence.

Generalized Periphrases. Based on the assumption that similar prefixes and suffixes are periphrased similarly, we train two other trie-based classifiers that learn the distribution of intermediate fillers per prefix and suffix. E.g. if we found the periphrasis "Hersteller von Aluminium" (manufacturer of aluminum) for the compound "Alminiumhersteller", then the prefix (suffix) classifier learns that "von" is likely to be used on the preceding (subsequent) position "Aluminium" ("hersteller") once.

Using this information, periphrases for arbitrary compound splits can be obtained by applying the classifiers on the parts. If intermediate filler information for both parts of a split is consistent, i.e. both classifiers overlap on the intermediate fillers, then the common filler is proposed which maximizes $\log(f_{\mathrm{p}1}+1) * \log(f_{\mathrm{p}2}+1)$ where $f_{\mathrm{p}i}$ is the number of occurrences of this filler with the part pi in the training data. In case of contradictory results, the most frequent filler is returned. Table 3 shows example periphrases and the training information for both prefix and suffix splits.

Table 3. Sample training set for the trie-based classifiers for intermediate fillers

Word	Periphrase	Right-position Trie	Right-position Class	Left-position Trie	Left-position Class
Aluminiumhersteller	Hersteller von Aluminium	Aluminium	von	Hersteller	von
Arbeitsaufwand	Aufwand bei der Arbeit	Arbeit	bei der	Aufwand	bei der

3 Experiments

3.1 Corpus Data

As corpus, we used the German corpus of Projekt Deutscher Wortschatz which comprises almost 1 billion tokens in about 50 million sentences [9]. The corpus was tokenized and preprocessed as described in the previous section.

3.2 Evaluation Data

Unfortunately, there is no publicly available standard dataset for German compound noun decomposition we are aware of. To judge the quality of our method, we created such a set, which will be available for download upon publication.

We report results on two evaluation sets: The first set is the CELEX database for German with a size of 90 077 splitted nouns [3]. Since here, all morphologically separable parts are marked as splits (also case endings etc.), we cannot hope for a high recall on this set. Nevertheless, we aim at high precision, since compound splits are marked (amongst many more splits). The second test set was created manually by the authors: it comprises 700 long German nouns of 14 different frequency bands from frequency = 1 up to frequency = 2^{13}. In this set, there are 13 words that are not compounds and should not be split, 640 items are compounds consisting of two pairs and 47 items consist of 3 parts - thus, we evaluate on 737 split positions.

Precision and Recall are defined as usual: Precision is obtained by dividing the number of correct splits by the number of splits taken by the method, recall is the number of correct splits divided by the total number of splits in the evaluation set. We further report results in the F1 measure, which is the harmonic mean of Precision and Recall.

3.3 Results

In our experiments we used parametrizations with the following values: mM = 4 (which has shown to be a good choice in premliminary experiments), mFr = 1, 2, 3, mTFr = 1, 2, 5, Fr, and d = 1, 2, 3. Here, mTFr = Fr means, that every identified part of a compound has to be at least as frequent as the whole compound candidate.

Splits. The counts of the splitting candidates are shown in Tab. 4 and 5. Table 4 shows the total numbers of splitting candidates after the search of the parts in the corpus and the number after filtering the best of different splits for the same compound by computing the geometric mean of the part frequencies (cf. [5]). Table 5 shows how many of these candidates could be found in the gold standard sets.

Tables 6 and 7 demonstrate that a higher threshold on minimum part frequency (mTFr) leads to higher Precision, since e.g. typing errors and spurious words from the word list are excluded. mTFr = Fr is not a viable option, because the most compound candidates are very low frequent so that the compound parts can also be very low frequent and thus spurious. However, this gain in Precision is traded of with a low Recall, as Tables 4 and 5 indicate.

Table 4. Total number of candidate splits

mM	mFr	mTFr							
		1		2		5		Fr	
		total	best	total	best	total	best	total	best
4	1	3114058	2490633	1554977	1405486	685604	653691	2051581	1710628
	2	1460443	1147076	719961	648802	309876	295624	397966	367071
	3	1013859	789987	496302	447016	211482	201983	174307	165285

Table 5. The number of candidate splits found in the gold standard sets

mM	mFr	mTFr							
		1		2		5		Fr	
		700	CELEX	700	CELEX	700	CELEX	700	CELEX
4	1	642	35948	362	19387	162	8094	244	11812
	2	474	28244	271	15252	129	6357	76	4108
	3	436	25230	249	13602	121	5625	54	2458

Table 6. Evaluation of candidate splits against the 700 manually splitted nouns

mM	mFr	mTFr											
		1			2			5			Fr		
		prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1
4	1	0.84	0.73	0.78	0.87	0.43	0.58	0.87	0.19	0.31	0.81	0.27	0.40
	2	0.85	0.54	0.66	0.86	0.31	0.46	0.85	0.15	0.25	0.82	0.08	0.15
	3	0.86	0.51	0.64	0.86	0.29	0.44	0.86	0.14	0.24	0.83	0.06	0.11

Table 7. Evaluation of candidate splits against CELEX

mM	mFr	mTFr											
		1			2			5			Fr		
		prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1
4	1	0.60	0.16	0.25	0.64	0.09	0.16	0.71	0.04	0.08	0.57	0.05	0.09
	2	0.63	0.13	0.21	0.66	0.07	0.13	0.73	0.03	0.06	0.69	0.02	0.04
	3	0.64	0.12	0.19	0.67	0.07	0.12	0.73	0.03	0.06	0.75	0.01	0.03

Table 8. Evaluation of generalized splittings against the 700 manually splitted nouns

mM	mFr	mTFr											
		1			2			5			Fr		
		prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1
4	1	0.54	0.59	0.57	0.52	0.49	0.5	0.44	0.36	0.39	0.53	0.54	0.54
	2	0.58	0.66	0.61	0.52	0.51	0.52	0.42	0.36	0.39	0.53	0.48	0.5
	3	0.56	0.68	0.62	0.5	0.52	0.51	0.41	0.37	0.39	0.5	0.43	0.46

Nor surprising, using the generalized splitter increases recall (compare figures in Tab. 6 with Tab. 8 and Tab. 7 with Tab. 9).

In summary, if one aims at high-quality splits but it is not required that all compounds get splitted, then a restrictive filter on candidate splits should be preferred. If the objective is to maximize F1 as Precision-Recall tradeoff, then the generalized splitter is the option to pursue.

Periphrases. To our knowledge, this is the first research aiming at automatically constructing periphrases for noun compounds from corpora. Thus, we have to manually evaluate our findings. A periphrasis was counted as correct if it

Table 9. Evaluation of generalized splittings against CELEX

mM	mFr	mTFr											
		1			2			5			Fr		
		prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1
4	1	0.50	0.21	0.29	0.47	0.17	0.25	0.44	0.13	0.20	0.47	0.17	0.25
	2	0.52	0.23	0.31	0.49	0.18	0.26	0.42	0.14	0.21	0.49	0.16	0.24
	3	0.52	0.23	0.32	0.48	0.18	0.26	0.41	0.14	0.20	0.46	0.15	0.23

was a grammatical noun phrase and its interpretation matches the default interpretation of the compound. E.g. "Kameraleute" (camera crew) is correctly periphrased as "Leute hinter der Kamera" (people behind the camera). "Leute vor der Kamera" (people in front of the camera) would be semantically wrong, "Leute zwischen Kamera" (people between camera) is not a grammatical noun phrase.

When taking the gold standard splits from our reference set of 700 words, our program gathered 216 periphrase candidates from the corpus, of which 160 were correct – a precision of 74%, recall at 23%, F1 at 35%. Using the generalized periphraser on the gold standard splits yielded 267 correct and 433 incorrect periphrases. We observed that some of the periphrases were correct as candidates and wrong for the generalized periphraser, so we propose the following setup:

1. If a candidate periphrase is found in the corpus, then accept it.
2. Otherwise, apply the generalized periphraser.

This experiment resulted in 336 correct and 364 incorrect periphrases (Precision = Recall = F1 = 336/700 = 48%).

4 Conclusions and Further Work

We discussed several ways to approach the problem of long one-word-compounds for some languages, which causes a high OOV rate in numerous NLP applications. First, we discussed how to split compounds into their respective parts: Similar to previous approaches like [8], we extract candidate splits by checking possible concatenations of short words against the long compounds, and rank several possible splits according to the geometric mean of the parts' frequencies as in [5] and propose to filter the candidate list according to criteria on frequency of parts and compounds. Since good precision values can only be obtained in hand with low recall, we propose to build a generalized splitter, taking the candidate splits as training. In this way, we increase overall F1. In a second experiment, we aim at periphrasing compounds to resolve their semantics. For this, we search our corpus for short snippets starting and ending with compound parts and having only few intermediate stopword fillers. Here, we are able to extract periphrases for our evaluation set with 74% precision and 23% recall (F1 = 35%). In a similar setup as in the splitting experiments, we train a generalized periphraser from all periphrases found in the corpus, again improving on F1 up to 48% by increasing recall and some loss in precision.

This is, to our knowledge, the first attempt to finding periphrases for compounds from corpora in a completely unsupervised and knowledge-free fashion. Our work can serve as a baseline for further experiments in this direction. For further work, we propose the following:

- Checking the splits according to existence of corpus periphrasis and using only splits that yielded a periphrasis for training of the generalized splitter. This could increase the quality of the generalized splits.
- Extenstion to *Fugenelemente*: Many compounds in Germanic languages contain the letter "s" between parts for reasons of easier pronounciation, e.g. in "Prüfung*s*vorbereitung*s*stress". Until now, this language feature is ignored. A possibility would be to explicitely provide a list of these very few fillers; however, more interesting would be to find them automatically as well.
- Evaluation of different filters for candidate splits and different measures for periphrasis selection
- Experiments for other languages, incuding more compounding languages, but also non-compounding ones for sanity-check.

References

1. Braschler, M., Ripplinger, B.: Stemming and decompounding for german text retrieval. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 177–192. Springer, Heidelberg (2003)
2. Brown, R.D.: Corpus-driven splitting of compound words. In: Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI) (2002)
3. Burnage, G., Harald Baayen, R., Piepenbrock, R., van Rijn, H.: CELEX: a guide for users. CELEX (1990)
4. Finkler, W., Neumann, G.: Morphix. a fast realization of a classification-based approach to morphology. In: 4. Österreichische Artificial-Intelligence-Tagung. Wiener Workshop - Wissensbasierte Sprachverarbeitung (1998)
5. Koehn, P., Knight, K.: Empirical methods for compound splitting. In: Proceedings of EACL, Budapest, Hungary, pp. 187–193 (2003)
6. Langer, S.: Zur Morphologie und Semantik von Nominalkomposita. In: Tagungsband der 4. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS) (1998)
7. Larson, M., Willett, D., Köhler, J., Rigoll, G.: Compound splitting and lexical unit recombination for improved performance of a speech recognition system for german parliamentary speeches. In: Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP) (2000)
8. Monz, C., de Rijke, M.: Shallow morphological analysis in monolingual information retrieval for dutch, german, and italian. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (eds.) CLEF 2001. LNCS, vol. 2406, pp. 262–277. Springer, Heidelberg (2002)
9. Quasthoff, U., Richter, M., Biemann, C.: Corpus portal for search in monolingual corpora. In: Proceedings of the LREC (2006)
10. Schiller, A.: German compound analysis with wfsc. In: Yli-Jyrä, A., Karttunen, L., Karhumäki, J. (eds.) FSMNLP 2005. LNCS (LNAI), vol. 4002, Springer, Heidelberg (2006)

11. Sjöbergh, J., Kann, V.: Finding the correct interpretation of swedish compounds – a statistical approach. In: Proceedings of LREC, Lisbon, Portugal (2004)
12. Turney, P.D.: Expressing implicit semantic relations without supervision. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (Coling/ACL-06), Sydney, Australia, pp. 313–320 (2006)
13. Witschel, F., Biemann, C.: Rigorous dimensionality reduction through linguistically motivated feature selection for text categorisation. In: Proceedings of NODALIDA (2005)
14. Yun, B.-H., Lee, H., Rim, H.-C.: Analysis of korean compound nouns using statistical information. In: Proceedings of the 22nd Korea Information Science Society Spring Conference (1994)