Noun Compound and Named Entity Recognition and their Usability in Keyphrase Extraction

István Nagy T.[1], **Gábor Berend**[1] and **Veronika Vincze**[2]
[1]Department of Informatics, University of Szeged
{nistvan,berendg}@inf.u-szeged.hu
[2]Hungarian Academy of Sciences, Research Group on Artificial Intelligence
vinczev@inf.u-szeged.hu

Abstract

We investigate how the automatic identification of noun compounds and named entities can contribute to keyphrase extraction and we also show how previously identified noun compounds affect named entity recognition and vice versa, how noun compound detection is supported by identified named entities. Our experiments demonstrate that already known noun compounds yield better performance in named entity recognition and already known named entities enhance noun compound detection. The integration of noun compound and named entity related features into a keyphrase extractor also proves to be more effective than the model not including them. Our results indicate that the above features tend to be beneficial in several NLP-related tasks.

1 Introduction

In natural language processing, the proper treatment of multiword expressions (MWEs) is essential for many higher-level applications (e.g. information extraction or machine translation). Multiword expressions are lexical items that can be decomposed into single words and display idiosyncratic features (Sag et al., 2002), in other words, they are lexical items that contain space. They are frequent in language use and usually exhibit unique and idiosyncratic behavior, thus, they often pose a problem to NLP systems. Named entities (NEs) are another class of linguistic elements that require special treatment in many NLP systems ranging from information retrieval to machine translation.

In this paper, we demonstrate how the automatic identification of noun compounds and named entities can contribute to keyphrase extraction and we also investigate how previously identified noun compounds affect named entity recognition (NER) and vice versa, how noun compound detection is supported by identified named entities. We briefly describe our methods, then discuss our results in detail. We argue that previous knowledge of noun compounds can enhance keyphrase extraction and NER while previously identified NEs can contribute to noun compound identification. We believe that employing NE- and noun compound-related features in other higher-level applications will also enhance performance.

2 Noun compounds and named entities in NLP applications

A compound is a lexical unit that consists of two or more elements that exist on their own. Compounds can be classified as follows (Sag et al., 2002; Kim, 2008): nominal compounds (*bass player*), adjectival compounds (*dark skinned*), adverbial compounds (*all in all*), prepositional compounds (*in front of*), and multiword conjunctions (*in order that*).

Named entity recognition is another widely researched topic in NLP. There are several methods developed for many languages and domains (Grishman and Sundheim, 1995; Chinchor, 1998; Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003). Multiword named entities can be composed of any words or even characters and their meaning cannot be traced back to their parts. For instance, *Ford Focus* refers to a car and has nothing to do with the original meaning of *ford* or *focus*, thus, it is justifiable to treat the whole expression as one unit.

Multiword expressions and named entities usually need special treatment in NLP systems due to their idiosyncratic features. Named entities often consist of more than one word, i.e. they can be seen as a specific type of multiword expressions / noun compounds (Jackendoff, 1997). The dis-

Proceedings of Recent Advances in Natural Language Processing, pages 162–169,
Hissar, Bulgaria, 12-14 September 2011.

tinction between noun compounds and multiword named entities is similar to that of between single-token common nouns and proper nouns. Although both noun compounds and multiword named entities consist of more than one word, they form one semantic unit and thus, they should be treated as one unit in NLP systems. Taking the example of POS-tagging, the linguistic behavior of compound nouns and multiword NEs is the same as that of single-word nouns, thus, they are preferably tagged as nouns (or proper nouns) even if the phrase itself does not contain any noun (e.g. *has-been* or *Die Hard*). Once identified as such, they can be treated similarly to single words in syntactic parsing for example.

However, the meaning of their parts and their connection alone cannot determine the semantics of the whole phrase, which yields that higher level applications need to pay special attention to them. For instance, in machine translation, it must be assured that the parts of a multiword expression are not translated separately, e.g. *racing car* should be translated to German as *Rennwagen*.

Noun compounds and multiword NEs behave similarly in language use in that both types function as one unit. It is this similarity that we would like to exploit when investigating the effect of already known NEs/noun compounds on the identification of the other type. On the other hand, our research focuses on the role of noun compounds and named entities in keyphrase extraction. In order to gain keyphrases from free texts, noun compounds might be of great help since once identified, they can be considered as one unit, i.e. like any other single word, which can be beneficial in e.g. frequency counts. Furthermore, the subject of texts is in many cases a named entity (in the Wiki50 corpus (Vincze et al., 2011), 39 articles are about a person, organization, location or another named entity), which fact underlines the importance of giving named entities a special treatment when identifying the topic of a text by keyphrases.

3 Experiments

For the evaluation of our models, we used Wiki50 (Vincze et al., 2011), in which several types of multiword expressions (including nominal compounds) and four classes of named entities were marked. Machine learning models were also evaluated on a 1000-sentence database from the British National Corpus that contains 345 noun compounds (Nicholson and Baldwin, 2008).

leave-one-out	**R**	**P**	**F**
MWE	58.07	69.86	63.42
MWE + NE	65.65	72.44	68.68
NE	85.58	86.02	85.81
NE + MWE	87.07	87.28	87.18

Table 1: Results of leave-one-out approaches in terms of precision (P), recall (R) and F-measure (F) in Wiki50. MWE: our CRF trained with basic feature set, which was extended with automatically collected MWE dictionary, MWE + NE: our CRF with MWE features extended with NEs as feature, NE: our CRF trained with basic feature set, NE + MWE: our CRF model with basic features extended with MWEs as feature.

3.1 Wikipedia based method for detecting noun compounds

For identifying noun compounds, we collected n-grams which occurred as links in English Wikipedia articles. Later, non-English terms, named entities and non-nominal compounds were automatically deleted from the list. We combined three methods: first, a noun compound candidate was marked if it occurred in the list. The second method involved the merge of two possible noun compounds: if *a b* and *b c* both occurred in the list, *a b c* was also accepted as a noun compound. Third, a noun compound candidate was marked if its POS-tag sequence matched one of the previously defined patterns. POS tags were determined by the Stanford POS Tagger (Toutanova and Manning, 2000). Results achieved by the combination of these methods are shown in the *DictCombined* row of Table 2.

3.2 Machine Learning approaches

In addition to the above-described approach, we defined another method for automatically identifying noun compounds. The Conditional Random Fields (CRF) classifier was used (MALLET implementations (McCallum, 2002)). The feature set includes the following categories (Szarvas et al., 2006):

orthographical features: capitalisation, word length, bit information about the word form (contains a digit or not, has uppercase character inside the word, etc.), character level bi/trigrams;

dictionaries of first names, company types, denominators of locations; noun compounds col-

lected from English Wikipedia (see 3.1);
frequency information: frequency of the token, the ratio of the token's capitalised and lowercase occurrences, the ratio of capitalised and sentence beginning frequencies of the token which was derived from the Gigaword dataset[1];
shallow linguistic information: part of speech;
contextual information: sentence position, trigger words (the most frequent and unambiguous tokens in a window around the word under investigation) from the train text, the word between quotes, etc.
To identify noun compounds we used the Wiki50 corpus to train CRF classification models (they were evaluated in a leave-one-document-out scheme). Results are shown in the *MWE* row of Table 1.

In order to use the Wiki50 corpus for testing only, we automatically generated a train database for the CRF trainer. The train set consists of 5,000 randomly selected Wikipedia pages and we ignored those containing lists, tables or other structured texts. Since this document set has not been manually annotated, dictionary based noun compound labeling was considered as the gold standard. As a result, we had a less accurate but much bigger training database. The CRF model was trained on the automatically generated train database with the above presented feature set. Results can be seen in *CRF* row of Table 2. However, the database included many sentences without any labeled noun compounds hence negative examples were overrepresented. Therefore, we thought it necessary to filter the sentences: only those with at least one noun compound label were retained in the database (*CRF + SF*). With this filtering methodology the CRF could build a better model. The above-described feature set was completed with the information that a token is a named entity or not. The *MWE + NE* row of Table 1 shows that this feature proved very effective in the leave-one-document-out scheme, so we used it in the automatically generated train database too. As shown in the *CRF + NE* row of Table 2, the CRF model which was trained on the automatic training set could achieve better results with this feature than the original *CRF*.

First, the Stanford NER model was used for identifying NEs. However, we assumed that a model trained on Wikipedia could more effectively identify NEs in Wikipedia (as it is the same domain). Therefore, we merged the four NE classes marked in Wiki50 into one NE class to train the CRF with common feature set described above. Results are shown in the *NE* row of Table 1. The *CRF + OwnNE + SF* row in Table 2 represents results achieved when we exploited as features the NEs that were identified by using the entire Wiki50 corpus as the training dataset. Although the *CRF + NE + SF* (when NEs were identified by the Stanford model) did not achieve better results than the *CRF + SF*, our Wikipedia based NE CRF model to identify NEs in the automatically generated training dataset (*CRF + OwnNE SF*) yielded better F-score than *CRF + SF*, which means that NE is a good feature in the identification of noun compounds. Since the sentence filtering yielded better results, in the following this approach will be used.

Sometimes it was not unequivocal to decide whether a multiword unit is a noun compound or a NE (e.g. *Attorney General*): some of the dissimilarities between the manual annotations were related to this problem. However, we assumed that a term can occur either as a NE or a noun compound. Therefore, if the dictionary method marked a particular word as noun compound and the NE model also marked it as NE, we had to decide which mark to delete. The *CRF + OwnNELeft + SF* row in Table 2 shows results we achieved if the NE labeling was selected as feature and the standard noun compound notation was removed, whereas the row *CRF + MWELeft + SF* refers to the scenario when the NE feature was deleted, and the standard noun compound notation remained.

We also wanted to see what results the above described approaches can achieve in another corpus. So we evaluated our methods on the BNC dataset too, these results are shown in Table 3. In Table 3 it can be seen that our approaches achieve worse results on the BNC dataset than on Wikipedia. This is largely due to the fact that our approaches rely heavily on Wikipedia. In addition, there are differences between the two corpora. For example, in the BNC dataset only compounds with two parts are marked while in the Wikipedia corpus noun compounds with 3 or more tokens can also occur. Due to this, the method of merging overlapping noun compounds could not even be used here. However, the difference between the CRF-

[1]Linguistic Data Consortium (LDC), catalogId: LDC2003T05

Approach	R	P	F	R	P	F
mwetoolkit	-	-	-	12.41	38.32	18.75
DictCombined	52.47	59.45	55.75	50.10	60.46	54.81
CRF	44.38	58.42	50.44	43.69	60.10	50.60
CRF + SF	53.39	56.66	54.98	52.94	57.57	55.15
CRF + NE	45.81	58.37	51.33	45.16	59.84	51.48
CRF + NE + SF	53.12	55.89	54.47	52.72	57.26	54.90
CRF + OwnNE + SF	53.29	57.60	55.36	52.84	59.8	56.13
CRF + OwnNELeft + SF	53.44	57.60	55.44	53.32	59.81	56.38
CRF + MWELeft + SF	53.53	58.74	56.02	53.01	59.67	56.14

Table 2: Results of different methods for noun compounds in terms of precision (P), recall (R) and F-measure (F) in Wikipedia corpus. mwetoolkit: the mwetoolkit system, DictCombined: combination of dictionary based methods, CRF: our CRF model trained on automatically generated database, SF: sentences without any MWE label filtered, NE: NEs marked by Stanford NER used as feature, OwnNE: NEs marked by our CRF model (trained on Wikipedia) used as feature, OwnNELeft: the NE labeling selected as feature and the standard noun compound notation removed, MWELeft: the NE feature deleted and the standard noun compound notation selected.

based and dictionary-based approaches is bigger in the BNC dataset. Furthermore, in this corpus too, CRF approaches enhanced with the NE feature performed best.

We found only one available other system to English noun compound recognition. This is the mwetoolkit system (Ramisch et al., 2010), a language-independent tool developed for collecting MWEs from texts (which is able to identify noun compounds). We evaluated it on these two corpora too. This system also relies heavily on POS tag features, therefore we completed the mwetoolkit POS tag rules with our POS rules. However, the mwetoolkit basically does not mark MWEs in the raw text, it just extracts noun compounds from the text, i.e. multiple occurrences of the same MWE are not taken into account. Therefore, in order to compare the results of our approaches to those of mwetoolkit, we assessed our methods similarly to the evaluation scheme used in the mwetoolkit. The results of mwetoolkit and our methods on the Wikipedia corpus can be seen on the right side in Table 2 and the BNC dataset on the right site in Table 3. As the tables show, with this evaluation method we achieve better F scores. This is probably due to that if a particular phrase occurs several times in the text and we cannot identify it, it counts as only one recall error in this evaluation, and in the other evaluation, each occurrence of the same MWE must be identified. The right handside of Tables 2 and 3 shows that we were able to achieve considerably better results than mwetoolkit. Again, in this type of evaluation, CRF models which used NEs as feature reached the best F-score. The mwetoolkit style evaluation is useful in e.g. collecting dictionary entries while the other type of evaluation is useful in e.g. information extraction or machine translation.

3.3 Named Entity Recognition with MWEs

As explained above, NEs are good features when we would like to extract noun compounds from texts. Therefore, we investigated the usability of noun compounds in named entity recognition. So we used the Wiki50 corpus to train CRF classification models with the basic feature set, which was extended with the feature noun compound MWE for NE recognition and they were evaluated in a leave-one-document-out scheme. Results of these approaches are shown in the *NE + MWE* row of Table 1. Comparing these results to those of the *NE* method (when the CRF was trained without the noun compound feature), noun compounds are also beneficial in NE detection.

4 Keyphrase extraction

Keyphrase extraction aims at the determination of the most important phrases of documents. The domain of keyphrase extraction most frequently involves scientific literature, but there have been other works that deal with other genres of texts as well (such as news articles as done in Farkas et al.

Approach	R	P	F	R	P	F
mwetoolkit	-	-	-	10.22	18.84	13.26
DictCombined	30.39	37.13	33.42	31.31	42.25	35.97
CRF	27.27	40.49	32.59	30.44	42.20	35.37
CRF + SF	34.91	39.48	37.06	39.11	41.33	40.19
CRF + NE	27.27	38.70	31.99	30.44	40.88	34.89
CRF + NE + SF	31.97	40.73	35.83	38.64	43.65	40.99
CRF + OwnNE + SF	36.78	36.10	36.43	41.22	37.93	39.50
CRF + NELeft	40.28	39.35	39.81	44.68	40.29	42.37
CRF + MWELeft	36.57	40.60	38.48	40.98	42.68	41.81

Table 3: Results of different methods for noun compounds in terms of precision (P), recall (R) and F-measure (F) in BNC dataset. mwetoolkit: the mwetoolkit system, DictCombined: combination of dictionary based methods, CRF: our CRF model trained on automatically generated database, SF: sentences without any MWE label filtered, NE: NEs marked by Stanford NER used as feature, OwnNE: NEs marked by our CRF model (trained on Wikipedia) used as feature, OwnNELeft: the NE labeling selected as feature and the standard noun compound notation removed, MWELeft: the NE feature deleted and the standard noun compound notation selected.

(2010)). Since keyphrases can be interpreted as the most important phrases of a document with respect to its content, their utilization in various NLP systems – ranging from document summarization to information retrieval or document classification – can be beneficial.

The fact that MWEs often prove to be proper keyphrases as well implies that the knowledge of MWEs in a given text can be exploited in the determination of the keyphrases of that document. However, we note that the two tasks (i.e. finding the MWEs and the keyphrases of documents) should be treated differently, since not all multiword expressions behave necessarily as keyphrases in all environments (e.g. although the phrase *research group* is definitely an MWE, its treatment as a keyphrase when it is present in the affiliations part of a scientific paper is not likely to be a valid choice for such a phrase that describes well the content of the document.)

In order to examine the possible utility of the usage of multiword expressions in the task of keyphrase extraction, we conducted experiments in this field. In our experiments we regarded the extraction of keyphrases from scientific documents as a supervised learning task, similarly to others (Frank et al., 1999; Turney, 2003; Witten et al., 1999). As for the dataset of our experiments, we used that of the shared task on keyphrase extraction of SemEval-2 (Kim et al., 2010).

The dataset is a subset of the ACM Digital Library and consists of 244 scientific publications of length ranging from 6 to 8 pages from four different research areas in computer science and economics. The documents were split into a training set of 144 documents and a test set of 100 documents by the organizers of the shared task. For training and testing our system, we used the keyphrases assigned to the documents coming from the readers of the papers of the dataset (similarly as it was done at the shared task).

4.1 Methodology

In our system we used the supervised learning approach for keyphrase extraction, in which the keyphrases of a document are determined by first identifying a set of potentially good phrases, then classifying its elements as either proper or non-proper keyphrases, based on the prediction of a machine learned model. We used the machine learning framework of MALLET (McCallum, 2002) for learning the proper keyphrases. Experiments using Maximum Entropy and Naïve Bayes classifiers were both conducted.

One key aspect in keyphrase extraction is the way keyphrase nominates are selected and represented. As the number of potentially extracted n-grams and that of genuine keyphrases among them shows high imbalancedness usually, keyphrase nominates are worth to be filtered, instead of using any successive n-grams. In our definition keyphrase candidates were n-

grams that were not longer than 4 tokens and started with a non-stopword token having either a `noun`, `adjective` or `verb` POS-code. Finally, an n-gram to be regarded as a keyphrase aspirant was also required to end with a non-stopword token having a POS-code either `noun` or `adjective`. Some phrases that fulfilled the above mentioned criteria were still discarded, due to positional rules, e.g. no phrase was regarded as a keyphrase aspirant if it occurred only in the *References* part of an article. This way 39,838 phrases were extracted from the 144 documents of the training corpus, which served as our training examples.

Once we had the keyphrase candidates, they had to be brought to a normalized form. The normalization of an n-gram consisted of lowercasing and Porter-stemming each of the lemmatized forms of its tokens, then putting these stems into alphabetical order (while omitting the stems of stopword tokens). With this kind of representation it was then possible to handle two syntactically different, but semantically equivalent phrases, such as *diffusion of innovation* and *Innovation diffusion* in the same way. For the linguistic analysis of the articles (i.e. tokenizing, lemmatization, POS-tagging) we used the Stanford CoreNLP API [2].

As for a baseline for our systems, we tried out KEA (Witten et al., 1999) as one of the most cited supervised keyphrase extracting tool, and also implemented its features in our system, which has its own strategy for generating keyphrase aspirants but uses the same standard features as well and uses the machine learning framework of MALLET. The two basic features for the keyphrase extraction system in KEA are the **tf-idf** score for an n-gram and **its relative first occurrence** within its context (i.e. the quotient of the first position of a certain n-gram and the length of the whole containing document).

To show the added value of MWEs in the task of keyphrase extraction, we designed a feature that indicated whether a certain n-gram (1) is an MWE, (2) can be built up from more MWEs, or just simply is the (3) superstring of at least one MWE. In order to do this we constructed a wide list of MWEs from Wikipedia (dump file 2011-01-07): we gathered all the links and formatted (i.e. bold or italic) text on Wikipedia that was at least two tokens in length, started with lowercase letters and contained only English characters or some punctuation. Based on this list, an alignment of its elements and the corpus was carried out (taking care of linguistic alternations), regarding those n-grams as genuine MWEs that started and ended with tokens of either a `noun` or `adjective` POS-code and had no other (possibly zero) tokens in between them that were of POS-code either `noun`, `adjective`, `preposition` or `possessive ending`. Thus when deciding on the MWE-related features of a keyphrase aspirant, we only had to decide if it was (1) annotated by our automatic process (taking the MWE list extracted from Wikipedia and the POS-sequence of a candidate into account) as an MWE in its full length (e.g. *maximal social welfare ratio*); (2) said to be able to put together from two MWEs present in our list (e.g. *resource allocation problems*, where *resource allocation* and *allocation problems* were in our list separately, but not as one phrase); (3) said to be a superstring of at least one MWE (e.g. *general analysis remains*, due to the presence of *general analysis*). Results achieved by KEA and our system (with and without using the above mentioned MWE-feature) are present in Table 4.

Besides the utilization of MWEs in the keyphrase extraction task, we were also interested in the effect of using features involving named entities. In order to investigate this, we implemented a set of binary features that were related to the orthography and semantics of keyphrase aspirants, as NEs usually both have special orthographic characteristics and special semantic roles in their content. For the determination of these feature values, we assigned the NE annotation of Stanford CoreNLP to keyphrase aspirants in such a manner that the feature values set to be true also implied the positions of the tokens having a specific NE-class within the keyphrase candidate. The position of one token of an n-gram was incorporated into the feature space as follows: separate features were created to indicate if an n-gram contained a certain type of NE-class standing at the beginning (B), inside (I) or at the end (E) of a keyphrase candidate. We also reserved a symbol for single token (S) keyphrase aspirants. For instance, *Nash* got positive value for the feature *S-PER* whereas *Nash equilibrium* had the feature *B-PER* set as true (and S-PER as false, naturally).

Strange orthography also had its binary features for n-grams incorporating similarly the position

[2] http://nlp.stanford.edu/software/corenlp.shtml

	Naïve Bayes			Maximum Entropy		
	Top-5	Top-10	Top-15	Top-5	Top-10	Top-15
KEA	22.2/9.23/13.04	18.0/14.96/16.34	15.53/19.37/17.24	20.4/8.48/11.98	18.2/15.13/16.52	15.93/19.87/17.68
BL	9.6/4.0/5.64	8.9/7.4/8.08	8.3/10.4/9.25	11.8/4.9/6.93	9.6/8.0/8.72	8.7/10.8/9.62
NE	7.6/3.2/4.46	5.7/4.7/5.17	5.2/6.5/5.77	14.4/6.0/8.46	10.9/9.1/9.9	10.1/12.6/11.25
MWE	18.4/7.6/10.8	13.7/11.4/12.44	11.1/13.8/12.28	18.4/7.6/10.8	14.4/12.0/13.07	10.9/13.6/12.13
COM.	12.6/5.2/7.4	12.1/10.1/10.99	10.0/12.5/11.1	13.8/5.7/8.1	14.8/12.3/13.44	13.2/16.5/14.65
EXT.	8.8/3.7/5.17	7.6/6.3/6.9	6.7/8.3/7.4	25.4/10.6/14.91	20.8/17.3/18.88	18.2/22.7/20.2
BEST	18.4/7.6/10.8	15.1/12.6/13.71	13.3/16.6/14.8	25.8/10.7/15.15	20.4/17.0/18.52	18.4/22.9/20.42
BMWE	21.6/9.0/12.68	17.3/14.4/15.71	14.4/18.0/15.98	26.0/10.8/15.27	21.2/17.6/19.25	19.0/23.7/21.09

Table 4: Evaluation results of keyphrase extraction in form of Precision/Recall/F-score at the top 5, 10 and 15 keyphrase levels using Naïve Bayes and Maximum Entropy classifiers. KEA: KEA system, BL: our baseline system using the standard KEA features, NE: our baseline system extended with the NE-related features, MWE: our baseline system extended with the MWE-related features, COM.: our baseline system extended with both NE- and MWE-related features, EXT.: extended feature set, BEST: the best combination of features without MWE-related features, BMWE: the best combination of features with MWE-related features

of the tokens that induced the feature to be set to true, e.g. in *UDDI registries* the feature *B-ORTHOGRAPHY* feature was set to true. A token was regarded to have strange orthography if it contained any uppercase letter besides its initial letter, or if it had more than 2 occurrences of the same character right after each other in any of its tokens. Results of the NE and orthography involving features are present in Table 4. To conclude our experiments we also experimented with the extension of the feature set that contained e.g. character suffix features, positional features within the document, POS-code related features, etc.

4.2 Results

As can be seen in Table 4, the Maximum Entropy models overperform the Naïve Bayes models. Best results are achieved for the top 15 keywords in each scenario. Results also show that the inclusion of the NE and MWE features proved useful in keyphrase extraction. Regarding NEs, although Naïve Bayes results somewhat declines when including NEs, its positive effect on the Maximum Entropy model is obvious. The addition of the MWE-features yielded better F-scores in each scenario, and best results can be achieved if all the useful features are enhanced by MWE-features, which clearly underlines the beneficiary effect of using MWEs in keyphrase extraction.

5 Discussion

Our results demonstrate that previously known noun compounds are beneficial in NER and identified NEs enhance MWE detection. This may be related to the fact that multiword NEs and noun compounds are similar from a linguistic point of view as discussed above – moreover, in some cases, it is not easy to determine even for humans whether a given sequence of words is a NE or a MWE (capitalized names of positions such as *Prime Minister* or taxonomic names, e.g. *Torrey Pine*). In the test databases, no unit was annotated as NE and MWE at the same time, thus, it was necessary to disambiguate cases which could be labeled by both the MWE and the NE systems. By fixing the label of such cases, disambiguity is eliminated, that is, the training data are less noisy, which leads to better overall results.

In keyphrase extraction, MWEs proved to be useful as well. This may be related to the fact that in many cases, keyphrases consist of multi-word tokens, thus, being an MWE might be suggestive of being a keyword aspirant too. It must be mentioned that not all MWEs are proper keywords, however, and must be filtered by other features as well. As for the importance of named entities in keyphrase extraction, in certain domains, person names tend to be common keyphrases (e.g. news) while in others, they do not typically function as keyphrases (e.g. biological publications), which highlights the domain-specificity of the problem. However, the keyphrase extractor can still profit from already known NEs: in one case, they can be excluded from the set of keyphrase aspirants while in the other case, they are proper keyword candidates.

6 Conclusions

In this paper, we investigated how the automatic identification of noun compounds and named en-

tities can contribute to keyphrase extraction and we also showed how previously identified noun compounds affect named entity recognition and vice versa, how noun compound detection is supported by identified named entities. Our experiments demonstrate that already known noun compounds yield better performance in NER and already known NEs enhance MWE detection. The integration of MWE- and NE-related features into a keyphrase extractor also proves to be more effective than the model not including them. Our results indicate that MWEs and NEs tend to be beneficial features in several NLP-related tasks. We firmly believe that our results in detecting noun compounds and named entities can be fruitfully applied in other higher-level applications as well in e.g. information extraction, document classification or machine translation.

Acknowledgments

This work was supported by the Project "TÁMOP-4.2.1/B-09/1/KONV-2010-0005 – Creating the Center of Excellence at the University of Szeged", supported by the European Union and co-financed by the European Regional Development Fund and by the project BELAMI financed by the National Innovation Office of the Hungarian government.

References

Nancy A. Chinchor. 1998. Overview of MUC-7/MET-2. In *Proceedings of MUC-7*.

Richárd Farkas, Gábor Berend, István Hegedűs, András Kárpáti, and Balázs Krich. 2010. Automatic free-text-tagging of online news archives. In *Proceeding of ECAI 2010*, pages 529–534, Amsterdam, The Netherlands. IOS Press.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proceeding of 16th IJCAI*, pages 668–673.

Ralph Grishman and Beth Sundheim. 1995. Design of the MUC-6 evaluation. In *Proceedings of the 6th Conference on Message Understanding*, pages 1–12, Stroudsburg, PA, USA. ACL.

Ray Jackendoff. 1997. *The Architecture of the Language Faculty*. MIT Press, Cambridge, MA.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of SemEval'10*, pages 21–26, Morristown, NJ, USA. ACL.

Su Nam Kim. 2008. *Statistical Modeling of Multiword Expressions*. Ph.D. thesis, University of Melbourne, Melbourne.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Jeremy Nicholson and Timothy Baldwin. 2008. Interpreting Compound Nominalisations. In *LREC 2008 Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 43–45, Marrakech, Morocco.

Carlos Ramisch, Aline Villavicencio, and Christian Boitet. 2010. Multiword Expressions in the wild? The mwetoolkit comes in handy. In *Coling 2010: Demonstrations*, pages 57–60, Beijing, China, August. Coling 2010 Organizing Committee.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. In *Proceedings of CICLing-2002*, pages 1–15, Mexico City, Mexico.

György Szarvas, Richárd Farkas, and András Kocsor. 2006. A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In *Discovery Science*, pages 267–278.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of EMNLP 2000*, pages 63–70, Stroudsburg, PA, USA. ACL.

Peter Turney. 2003. Coherent keyphrase extraction via web mining. In *Proceedings of IJCAI '03*, pages 434–439.

Veronika Vincze, István Nagy T., and Gábor Berend. 2011. Multiword expressions and named entities in the Wiki50 corpus. In *Proceedings of RANLP 2011*, Hissar, Bulgaria.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *ACM DL*, pages 254–255.