Multiword noun compound bracketing using Wikipedia

Caroline Barrière **Pierre André Ménard**
Centre de Recherche Informatique de Montréal (CRIM)
Montréal, QC, Canada
{caroline.barriere;pierre-andre.menard}@crim.ca

Abstract

This research suggests two contributions in relation to the multiword noun compound bracketing problem: first, demonstrate the usefulness of Wikipedia for the task, and second, present a novel bracketing method relying on a word association model. The intent of the association model is to represent combined evidence about the possibly lexical, relational or coordinate nature of links between all pairs of words within a compound. As for Wikipedia, it is promoted for its encyclopedic nature, meaning it describes terms and named entities, as well as for its size, large enough for corpus-based statistical analysis. Both types of information will be used in measuring evidence about lexical units, noun relations and noun coordinates in order to feed the association model in the bracketing algorithm. Using a gold standard of around 4800 multiword noun compounds, we show performances of 73% in a strict match evaluation, comparing favourably to results reported in the literature using unsupervised approaches.

1 Introduction

The noun compound bracketing task consists in determining related subgroups of nouns within a larger compound. For example (from Lauer (1995)), *(woman (aid worker))* requires a right-bracketing interpretation, contrarily to *((copper alloy) rod)* requiring a left-bracketing interpretation. When only three words are used, *n1 n2 n3*, bracketing is defined as a binary decision between grouping *(n1,n2)* or grouping *(n2,n3)*. Two models, described in early work by Lauer (1995), are commonly used to inform such decision: the adjacency model and the dependency model. The former compares probabilities (or more loosely, strength of association) of two alternative adjacent noun compounds, that of *n1 n2* and of *n2 n3*. The latter compares probabilities of two alternative dependencies, either between *n1* and *n3* or between *n2* and *n3*.

Most compound bracketing research has focused on three-noun compounds as described above. Some recent work (Pitler et al. (2010), Vadas and Curran (2007b)) looks at larger compounds, experimenting with a dataset created by Vadas and Curran (2007a) which we also use in our research. For larger noun compounds, the adjacency model alone will not allow longer range dependencies to be taken into account. This had been noted much earlier in Barker (1998) using examples such as *(wooden (((French (onion soup)) bowl) handle))* to show a long-range dependency between *wooden* and *handle*.

To allow for such long-range dependencies, our bracketing algorithm looks at all possible word associations within the full expression to make its decisions. The word associations are captured within an association model which goes beyond the adjacency and dependency models. The association model represents combined evidence about the possibly lexical, relational or coordinate nature of the links between all word pairs. In its current implementation, our association model relies on Wikipedia as a resource for obtaining all three types of evidence. Wikipedia is used in two forms: first as a list of terms and named entities (Wikipedia page titles), and second, as a large corpus obtained from the merging of all its pages. The resulting corpus is large enough to be used for statistical measures. The most current version contains 14,466,099 pages in English for an uncompressed file size of 47 gigabytes (including

Proceedings of the First Workshop on Computational Approaches to Compound Analysis, pages 72–80,
Dublin, Ireland, August 24 2014.

some metadata). To the best of our knowledge, no previous research has used Wikipedia for the noun bracketing task, and this research will explore its usefulness.

The reminder of this article will unfold as follows. Section 2 presents a brief literature review. Section 3 describes the dataset used in our experiments. Section 4 presents the bracketing algorithm, and Section 5 the implementation of a word association model using Wikipedia. Section 6 describes our evaluation approach, while results are presented and analysed in Section 7. Section 8 concludes and suggests future work.

2 Related work

Noun compound bracketing has not received as much attention as many other Natural Language Processing (NLP) tasks. Nakov and Hearst (2005) call it an understudied language analysis problem. Early work by Lauer (1995) took inspiration in even earlier linguistic work by Levi (1978). Lauer (1995) having devised a small dataset of 3-word noun compounds, his dataset was reused by various researchers (Lapata et al. (2004), Girju et al. (2005), Nakov and Hearst (2005)) who promoted the use of corpus-based empirical methods for the task.

To address the noun compound bracketing task, different authors use different datasets, different views on the problem (adjacency, dependency), different methods of resolution (supervised, unsupervised) and different constraints on the problem (compound seen in isolation or in context). Independently of such differences, all researchers have an interest in evaluating word-pair associations. Most recent research uses the Web for providing word pair association scores to their bracketing algorithm. The work of Lapata et al. (2004) shows usefulness of web counts for different tasks, including noun compound bracketing. The work of Pitler et al. (2010) intensively uses web-scale ngrams in a supervised task for large NP bracketing, showing that coverage impacts on accuracy. Beyond bigram counts on the web, varied and clever searches (Nakov and Hearst, 2005) have been suggested such as the use of paraphrases (*n1 causes n2*) or simpler possessive markers (*n1's n2*) or even the presence of an hyphen between words (*n1-n2*). All variations are to provide better word association estimates and improve bracketing. The use of web counts is sometimes complemented by the use of more structured resources, such as in Vadas and Curran (2007b) who combines web counts with features from Wordnet.

In our research, instead of web counts, we rely on a community-based encyclopedic resource, Wikipedia, for corpus-based evidence. We rely on the same resource to access a list of terms and entities. Although not much of the structure of Wikipedia is used in our current implementation, such as its categories or page links, we can envisage to use it in future work. Similarly to other researchers mentioned above, our goal is to gather evidence for word-pair association, although an important contribution of our work is to refine this notion of word-pair association into three subtypes of association: lexical, relational and coordinate. We suggest that a better characterization of the possible links among word pairs in a large compound will better inform the bracketing algorithm.

3 Dataset

Vadas and Curran (2007a) manually went through the Penn Treebank (Marcus et al., 1994) to further annotate large NPs. They openly published a *diff file* of the Penn Treebank to show their annotations which differ from the original. From this available file, we constructed our gold-standard dataset by extracting large NPs (three or more words) which only include relevant items (common and proper nouns, adverbs and adjectives), removing determiners, numbers, punctuations and conjunctions. The expressions were then verified for completeness, so that the opening bracket should be closed within the length of text defined in the differential file. Finally, tags and single words enclosing parentheses were removed to produce simplified versions of the bracketed expressions (e.g *(NML (NNP Nesbitt) (NNP Thomson) (NNP Deacon))* becomes *(Nesbitt (Thomson Deacon)))*.

Vadas and Curran (2007a) used a Named Entity annotator to suggest bracketing to the human annotators (who could accept or reject them). The entity types used were the ones defined by Weischedel and Ada Brunstein (2005) (e.g. Person, Facility, Organization, Nationality, Product, Event, etc). Named

entities could be kept *as-is* by the annotators or could be bracketed if deemed compositional. Annotators were also instructed to use a default right-bracketing (implicit in Penn Treebank) for difficult decision.

In our dataset, we transformed the ones left *as-is* into right-bracketed in order to have all expressions fully bracketed. This process might seem controversial, as it assumes compositionality of all named entities, which for sure, is a wrong hypothesis. The alternative, though, would require the bracketing algorithm to recognize named entities, which we consider outside the scope of this research. Furthermore, it would also be wrong to assume all named entities are non-compositional. For example *New York Stock Exchange* is clearly compositional, and a Named Entity Tagger based on Wikipedia would easily identify it as a named entity (although the use of Wikipedia as a source of named entities is also debatable). Clearly, no solution is satisfying. We opted for the approximation which provided a fully bracketed gold standard to which our results could be compared. We are aware that this will have a negative impact, in some cases, on our results.

The extraction produced a total 6,600 examples from which we removed duplicate expressions, yielding a corpus of 4,749 unique expressions. Among those unique expressions, 2,889 (60.95%) were three words long (e.g. *Mary Washington College*), 1,270 (26.79%) had four words (e.g. *standardized achievement tests scores*), 413 (8.71%) with five words (e.g. *annual gross domestic product growth*) and the remaining longer expressions (up to nine words) covered around 3.5% of the dataset[1].

4 Bracketing method

As in the work of Pitler et al. (2010), our bracketing algorithm takes into account all possible word pairs within the noun compound. This differs from Barker's algorithm Barker (1998) used in Vadas and Curran (2007b) which only uses local information, three-words at a time, in a right-to-left moving window. We briefly present our algorithm below and refer the reader to Ménard and Barrière (2014) for a more detailed explanation.

First, a list (L1) is created to contain every word pair that can be generated, in order, from an expression. For example, a list L1 {(A,B), (A,C), (A,D), (B,C), (B,D), (C,D)} would be created from expression "A B C D". Second, a dependency score needs to be assigned to each pair. Our bracketing algorithm actually builds a dependency tree and requires these dependency scores. We make the assumption that dependencies are implicitly directed left-to-right. This is an oversimplification, as there are a few cases, such as *Vitamin C* or *Cafe Vienna*, pointed in (Nakov, 2013), where the direction is reversed. Furthermore, this hypothesis is valid only for English and renders our algorithm less applicable to other languages. Although fair for English, this hypothesis should be revisited in future work.

The next step is building a final list of dependencies (L2) to represent the full dependency tree. To do so, the algorithm repeatedly selects from L1 the word pair with the maximum score and adds it to L2 only if both (a) the modifier has not already been used, and (b) the new pair does not create a crossing of modifier/head pairs in the expression. For example, if L2 already contains (AB)(C(DE)), then (BD) would create an invalid crossing and is not accepted. The selection of pairs from L1 ends when all words from the expression, except for the right-most one, are used as modifiers in L2.

Our algorithm is greedy and considers only the best score at every step. We have experimented with randomized greedy algorithms as well, choosing randomly between top N scores at each step, but since results did not improve, we do not report on them in the current article. The bracketing algorithm favours high dependency scores without consideration for the actual distance between word pairs in the source expression. This helps linking far reaching dependencies in noun compounds, but might also force some strong association between two distant words without regard to the soundness of using nearer words.

5 Implementing an association model using Wikipedia

Our association model contains three types of association: lexical, relational and coordinate. Each one will be measured using Wikipedia through different approximation strategies. The challenge is the integration of the association model with the bracketing algorithm. We mainly explore a solution of **score**

[1]We describe our dataset in more details in Ménard and Barrière (2014), and our extraction method is published as part of the LREC resources sharing effort.

modulation which does not require the bracketing algorithm to be modified but rather use the three association scores to modulate the dependency score required by the bracketing algorithm. We present below a basic dependency score, and then different strategies to transform the three types of association into modulation factors on that dependency score.

Basic dependency association: Based simply on the co-occurrence of two words in a corpus, this basic association will be influenced by the actual corpus (domain and size), and the association measure used. In our current experiment, Wikipedia pages are merged into a large corpus (47 Gigabytes) covering multiple domains. As for the association measure, we compare Dice and Point-Wise Mutual Information (PMI), although many more exist in the literature. Co-occurrence is not a direct measure of dependency, it is an approximation. A true dependency measure would require a syntactic analysis (using a link parser) of the whole corpus. We will explore this idea in future work.

Relational association: The relational association is a refinement to the dependency association. In semantic analysis of noun compounds, an important goal is to characterize the nature of the dependence between its words, such as cause, purpose, location, etc (see work by Girju et al. (2005), Nakov and Hearst (2005), Nastase et al. (2013) among many). Here, we do not require the identity relations, but rather search for indications of the relational status of a word pair. In our current implementation, relational association is naïvely determined by the presence of a preposition between two nouns. We use the prepositions: about, at, by, for, from, in, of, on, to, with. We search in the corpus for patterns such as "N1 *at* N2" and "N1 *for* N2", etc. The frequency of these will be used to boost the basic dependency association scores.

Coordinate association: Proximity sometimes refers implicitly to coordination, as for example the words *cotton* and *polyester* in the expression *cotton polyester shirt*. Explicit external evidence that these words often co-occur in a coordination relation could lower their dependency association in expressions such as *cotton polyester shirt*. To gather such evidence, we measure the frequency of explicit coordination between word pairs in Wikipedia. The common conjunctions: *or*, *and*, *nor* are used. We search in the corpus for patterns such as "N1 *or* N2" and "N1 *and* N2", etc. Contrarily to relational associations boosting the basic dependency association scores, coordinate associations should attenuate the dependency scores.

Lexical association: Based on the idea that many compounds, even named entities, are compositional, we want to determine the likeliness that a subexpression in a compound forms itself a lexical unit with a meaning of its own. To do so, we use a first approach requiring a set of corpus-based statistical approximations and a second approach requiring Wikipedia page titles.

- *Statistical approximation*: The presence of determiners (*a, an, the*) and plural forms are used as statistical evidence of lexical association. For example, starting with expression *cotton polyester shirt*, corpus analysis shows that *the cotton shirts* is frequent, which can be used to boost the dependency score between *cotton* and *shirt*. On the other hand, *the cotton polyesters* will be much less frequent. The presence of indicators (determiners and plurals) can be used independently, searching for patterns such as "*the* N1 N2" and "N1 plural(N2)", or together for patterns such as "*a* N1 plural(N2)".

- *Presence in Wikipedia*: A second strong indicator of lexical association for a word pair is its presence in an encyclopedic resource (Wikipedia). In fact, not only word pairs, but for any subcompound of two or more words are considered for look-up as Wikipedia entries. Since we now have lexical units of any length, rather than word pairs, our score modulation is not as straight forward. We thought of two different strategies.

  The first strategy, in line with score modulation, uses all word pairs found in the lexical units to boost dependency scores. For example, assuming the compound *ABCDE*, with *[BCD]* found as a lexical unit in Wikipedia. Then, the association scores of pairs *[BC],[CD],[BD]* are boosted equally (uniform boost). This will not help for any internal bracketing of *[BCD]*, but will reinforce the fact that *[BCD]* should stay together within the larger compound. A variant to uniform boost

Gold	Evaluated	Gold elements		Strict	Lenient	
		Subexpression	Binary tree		Subexpression	Binary tree
(a b) c	(a b) c	(a b)	a-b, b-c	100%	100%	100%
(a b) c	a (b c)	(a b)	a-b, b-c	0%	0%	50%
(a b) (c d)	(a b) (c d)	(a b), (c d)	a-b, c-d, b-d	100%	100%	100%
(a b) (c d)	a (b (c d))	(a b), (c d)	a-b, b-d, c-d	0%	50%	66.6%
(((a b) c) d) (e f)	a (b (c (d (e f))))	(a b), (a b c), (a b c d), (e f)	a-b, b-c, c-d, d-f, e-f	0%	25%	40%
			Average:	40%	55%	71.3%

Table 1: Applied examples of evaluation metrics.

is a right-attachment boost to mimic the default right bracketing in the gold standard for the longer units.

The second strategy is one of **compound segmentation**, in which lexical units found become segmentation constraints on the bracketing algorithm. Association scores are them measured between pairs of lexical units instead of between words pairs. We also try to minimize the number of entities within the compound. For example, assuming again we wish to bracket compound *ABCDE*, and find the possible three segmentations into lexical units using Wikipedia: (1)*[AB][CDE]*, (2) *[AB][CD][E]*, (3) *[ABC][DE]*. Only segmentations (1) and (3) are kept since they have two lexical units and not three. The association scores must then be calculated between pairs of lexical units, and within each lexical unit containing three words or more (to perform full bracketing). Bracketing within a lexical unit will be performed using the same bracketing methods described above. Bracketing between lexical units requires association scores between these units. For doing so, using the example above, we will search in corpus for cooccurrences of *[AB]* with *[CDE]* for segmentation (1), and *[ABC]* with *[DE]* for segmentation (3). Since statistics on longer units will be sparse in the corpus, we will also measure association scores between heads of the lexical units. For example, in segmentation (1) the association between heads *[B]* and *[E]* would be measured.

6 Evaluation metrics

Three methods are used to evaluate performances: strict, lenient binary tree and lenient sub-expression. The strict evaluation verifies that all bracketed groups of the gold-standard expression are exactly the same as those found in the evaluated expression, providing a score of 1 or 0. The two lenient evaluations compute the ratio between the number of matching groups from a gold expression with those found in the evaluated expression. In other words, lenient is the recall score based on the gold elements.

In lenient binary tree, each fully bracketed expression is parsed as a binary tree. From that tree, each modifier/head pair becomes a basic evaluation element. For example, in *(A (B C))*, two elements *A-C* and *B-C* are used for the evaluation process. This method boosts the performance level on most expressions, but especially those composed of three words, for which a minimum 50% is always obtained.

In lenient sub-expression, evaluation elements are rather sub-expressions to provide a more balanced score. The method extracts each bracketed group except the top-level group and removes all internal parentheses from each one. Thus, from the expression *(((A B) C) D)*, the method extracts *(A B)* and *(A B C)*. The two resulting sub-expressions become gold elements for comparison with those obtained from the evaluated expression. Table 1 shows five examples illustrating score variations using the different methods on expressions of different length.

7 Results

In section 5, we described various approaches to capture, using Wikipedia, the different types of association proposed in our model: lexical, relational and coordinate. We also presented two solutions for combining this more complex model with the bracketing algorithm of section 4 which expects a single type of association, that of dependency. Below, using a dataset of 4749 compound nouns, presented in section 3, we report on some interesting results.

Resource	Algorithm	Strict	Lenient
Wikipedia	Dice	55.00%	67.63%
	PMI	56.25%	68.98%
Google Web Ngram	Dice	51.80%	63.90%
	PMI	60.41%	72.47%

Table 2: Comparing basic association scores in Wikipedia and Google Web.

7.1 Baseline

To measure the impact of combining different types of associations, we first establish our baseline as the bracketing results obtained solely with the basic dependency association scores, as measured on Wikipedia. To further validate our baseline, we wish to compare it to the literature. The closest research providing comparable results on large compounds are Vadas and Curran (2007b) and Pitler et al. (2010), although both focus on supervised approaches, and furthermore, Vadas and Curran (2007b) use contextual features, assuming the noun compounds are to be bracketed in context. Still, Vadas and Curran (2007b) give some baseline results for an unsupervised approach (the supervised approach was promoted in their article) to which we compare our baseline. Far from an ideal comparison (which would be with the exact same dataset and setting), it still provides some indication of the performance of our baseline. They report exact match for complex NPs to be 54.66% for default right branching, 32.66% chi-square dependency and 35.86% chi-square adjacency. As we obtain around 55% for strict matches (see Table 2, first row), we seem above the unsupervised approach they used, which combined their association scores within an implementation of Barker's algorithm.

To confirm that merged Wikipedia pages form a large enough corpus in comparison to most recent work on noun bracketing using web counts (see section 2), we use the English Google Web Ngrams (Lin et al., 2010) (GWN), a 1T corpus contains n-gram counts collected from 1 trillion words of web text, and performed our bracketing algorithm with Wikipedia basic dependency scores, and GWN bigram scores. As shown in Table 2, results are comparable, slightly higher for Dice (55.0% compared to 51.8%) and slightly lower for PMI (56.25% compared to 60.41%).

Throughout our experiments, we have continued using both association measures (Dice and PMI), as well as performing both Barker's algorithm and our bracketing algorithm, but since our algorithm with Dice always gave better results (contrarily to the baseline in which PMI performed better), we only present those results in the following sections.

7.2 Corpus-based improvements

In Section 5, we described how the use of stop words (conjunctions, prepositions, determiners) combined with word pairs of interest could respectively modulate the basic dependency association scores to emphasize coordinate, relational, or lexical association.

For lexical association, word pairs preceded by determiners were searched for in the corpus. We tried different ways of combining association scores between the form with the determiner ("the N1 N2") and the word pair only (N1 N2), such as adding scores, keeping the maximum or minimum score. As well, we tried different ways of combining the scores obtained with the different determiners (a, the, an), again adding, keeping the maximum or the minimum score. Unfortunately, none of these variations helped. We also experimented with searching for plural forms in corpus to emphasize lexical association, which provided a small increase to the baseline as shown in Table 3.

For relational association, we searched for noun pairs with prepositions. The same merging strategies given above for the use of determiners we tried. The best configuration uses a relational boosting strategy of adding scores and a preposition merging strategy of using the minimum score among all prepositions. Even with the best combination, overall, the improvement is marginal as shown in Table 3.

For coordinate association, we searched for noun pairs with conjunctions. Similarly to determiners and prepositions, we tried different merging strategies. Since we are interested in an attenuation of the dependency score with the coordinate score, our merging strategies were of subtracting scores or using

Option	Strict	Lenient	Binary
Baseline	0.5500	0.6763	0.8132
Only including lexical association	0.5842	0.7106	0.8321
Only including relational association	0.5854	0.7093	0.8314
Only including coordinate association	0.5867	0.7110	0.8325

Table 3: Impact of corpus-based statistics (lexical, relational, coordinate association)

Option	Strict	Lenient	Binary
Baseline	0.5500	0.6763	0.8132
Using entity-based refinement (uniform distribution)	0.6020	0.7257	0.8408
Using entity-based compound segmentation	**0.7316**	**0.8213**	**0.8940**

Table 4: Use of entities

the minimum. Again, unfortunately, improvement is marginal, as shown in Table 3.

7.3 Entity-based improvements

Our second approach to promote the lexical unit association score is to find which sub-expressions of the compound are Wikipedia page titles. In Section 5, we suggested two strategies of using these entries, either **score modulation** or *compound segmentation*.

In score modulation, we tried uniform boosting and right boosting as explained in Section 5, with different boosting factors arbitrarily set between 10 and 100. The best result, obtained using a uniform boost with a factor of 50 is presented in Table 4. There is a small improvement using this method. The second strategy of compound segmentation is the one providing the most significant gain. An increase of 13% is obtained for the strict evaluation as shown in the last row of Table 4. For the sake of completeness, we reran all the different variations and parameters which are used for performing the within and between lexical units bracketing. The best configuration required that (1) basic dependency scores were actually replaced by scores obtained by finding plural forms in the corpus (lexical association), (2) determiners were not used, (3) the negative modulation from conjunctions (coordinate association) is obtained by subtracting their frequency from the basic scores, (4) the positive modulation of prepositions (relational association) is obtained by adding their frequency to the basic scores, (5) as different prepositions are searched in corpus, the one with minimum frequency should be taken to alter basic scores, same for conjunctions (6) the head of lexical units is used to measure the "between units" association scores.

7.4 Result analysis

We first note some aspects of the gold standard that would affect the adequacy of our algorithm, and our results.

- ***Noun compound status***: A few examples in the dataset contain very generic adjectives, such as: *(certain ((natural resource) assets))*, *(such ((gas management) contracts))*,*(most (structural engineers))*, or *((too much) attention)*. These are not problematic in themselves, but our statistical approximations for lexical, relational and coordinate associations are not adequate for these cases.

- ***Abbreviations***: Some examples in the gold standard contain abbreviations, for example, *(republican (u.s. sen.))*, *((american president) cos.)* or *(((el dorado) investment) co.)*. Again, these are not problematic in themselves, but we have not yet implemented anything in our algorithm to manage such cases.

- ***Ambiguity***: Some examples found in the gold standard, such as *((sun ((life assurance) society)) plc)* or *((magnetic (resonance imaging)) equipment)* are not obvious to us as being correct.

- ***Compositional examples***: On the positive side, the dataset certainly contains many interesting examples, such as *((new england) ((medical center) hospitals))*, *((northern california) (home prices))*,

(world-wide ((advanced materials) operations)), *(((lone star) spokesman) (michael london))*, or *((magnetic (resonance imaging)) equipment)*. These examples are interesting because they show a variety of right and left bracketing needed and a variety of named entities and terms of different compositional nature. Research on compound bracketing is required for those examples, as they will probably never end-up in even the most extensive lists of terms and named entities.

To better understand this dataset and the adequacy of our algorithm to its content, we intend, in future work, to perform a manual sampling to determine the types of compounds, and the possible ambiguities.

As for Wikipedia as a resource, it is very valuable and contains many named entities (places, corporations, persons, etc), but it can never contain all entities. For example, we will find *tadeusz mazowiecki* to help in bracketing *(polish (prime minister)) (tadeusz mazowiecki)*, but we will not find *bruno lucisano*, and wrongly bracket *(((rome (film producer)) bruno) lucisano)*.

Independently of the gold standard and the resource used, our method has multiple limitations and peculiarities. We believe that the general approach presented in this research is quite valid: a proposal for the refinement of generic association scores into three subtypes of associations: lexical, relational and coordinate associations. Nevertheless, the statistical approximations used for evaluating the different association types should be revisited and refined.

8 Conclusion

Although bracketing of three-word expressions has been performed quite successfully using unsupervised approaches with web-corpus resources ((Nakov and Hearst, 2005), (Vadas and Curran, 2007b)), compound bracketing of large expressions remains a challenge.

One research direction, taken by Vadas and Curran (2007b) and Pitler et al. (2010) is to investigate supervised learning approaches which will be able to build on the redundancy within the dataset. We take a different direction, that of developing a more complex association model and exploring Wikipedia in an unsupervised manner. Our research presents a noun compound bracketing algorithm which goes beyond the adjacency / dependency models presented so far in the literature. We suggest a method that takes into account different meaning of the proximity of two words, that of being part of the same lexical unit, or being coordinates, or being in a relation.

Our current implementation of our association model certainly provides improvement on the basic association scores, but it does not give a clear view of whether our corpus-based approximations are correct or not. This deserves future investigation into how to best approximate with statistical measures the notions of relational, coordinate and lexical associations. On the other hand, the use of Wikipedia as an encyclopedic resource to help determine lexical units certainly provides the most gain and the best results. On the dataset of 4749 compounds, our best results are 73.16% strict, 82.13% lenient and 89.40% binary tree evaluation. Further use of the structure of Wikipedia can be investigated to help characterize the different types of associations.

An important future goal is to refine the association model, and better anchor it in both linguistic and computational linguistic traditions of noun compound analysis. The model deserves to be studied in its own, regardless of its implementation, which here was performed using Wikipedia. A better understanding of the model and its impact on noun compound bracketing might direct us to better choices for the implementation of the association measures.

Lastly, similarly to other researchers who look at noun compound bracketing as the first step of semantic analysis of NPs to elicit semantic relations (purpose, cause, location, etc) between subgroups of words (Girju et al. (2005), Nastase et al. (2013)), we want to pursue our work into a more fine-grained understanding of noun compounds (Nakov, 2013), combining bracketing with the identification of specific noun relations.

9 Acknowledgements

This research project is partly funded by an NSERC grant RDCPJ417968-11, titled Toward a second generation of an automatic product coding system.

References

Ken Barker. 1998. A Trainable Bracketer for Noun Modifiers. In *Twelfth Canadian Conference on Artificial Intelligence (LNAI 1418).*

Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. 2005. On the semantics of noun compounds. *Computer Speech & Language*, 19(4):479–496, October.

Mirella Lapata, Portobello St, S Sheffield, and Frank Keller. 2004. The Web as a Baseline : Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP Tasks. In *Proceedings of the HLT-NAACL*, pages 121–128.

Mark Lauer. 1995. Corpus statistics meet the noun compound: some empirical results. *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 47–54.

Judith Levi. 1978. *The syntax and semantics of complex nominals*.

D Lin, KW Church, H Ji, and S Sekine. 2010. New Tools for Web-Scale N-grams. *LREC*.

Mitchell P Marcus, Santorini Beatrice, and Mary A Marcinkiewicz. 1994. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

Pierre André Ménard and Caroline Barrière. 2014. Linked Open Data and Web Corpus Data for noun compound bracketing. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 702–709, Reykjavik, Iceland.

Preslav Nakov and M Hearst. 2005. Search engine statistics beyond the n-gram: Application to noun compound bracketing. *Proceedings of the Ninth Conference on Computational Natural Language Learning*, (June):17–24.

Preslav Nakov. 2013. On the interpretation of noun compounds: Syntax, semantics, and entailment. *Natural Language Engineering*, 19(03):291–330, May.

Vivi Nastase, Preslav Nakov, Diarmuid O Seaghdha, and Stan Szpakowicz. 2013. *Semantic Relations Between Nominals*. Morgan and Claypool Publishers.

Emily Pitler, Shane Bergsma, Dekang Lin, and Kenneth Church. 2010. Using web-scale N-grams to improve base NP parsing performance. *Proceedings of the 23rd International Conference on Computational Linguistics*, (August):886–894.

David Vadas and JR Curran. 2007a. Adding noun phrase structure to the Penn Treebank. *45th Annual Meeting of the Association of Computational Linguistics*, (June):240–247.

David Vadas and JR Curran. 2007b. Large-scale supervised models for noun phrase bracketing. *10th Conference of the Pacific Association for Computational Linguistics*, (2004):104–112.

Ralph Weischedel and Ada Brunstein. 2005. BBN pronoun coreference and entity type corpus. Technical report, Linguistic Data Consortium.