Linked Open Data and Web Corpus Data for noun compound bracketing

Pierre André Ménard, Caroline Barrière
Centre de recherche en informatique de Montréal
405, avenue Ogilvy, bureau 101
Montréal (Québec), Canada
pierre-andre.menard@crim.ca, caroline.barriere@crim.ca

Abstract

This research provides a comparison of a linked open data resource (DBpedia) and web corpus data resources (Google Web Ngrams and Google Books Ngrams) for noun compound bracketing. Large corpus statistical analysis has often been used for noun compound bracketing, and our goal is to introduce a linked open data (LOD) resource for such task. We show its particularities and its performance on the task. Results obtained on resources tested individually are promising, showing a potential for DBpedia to be included in future hybrid systems.

Keywords: noun compound bracketing, linked open data, Google Ngrams

1. Introduction

In the field of computational linguistics, large corpora have been shown to be quite good for the task of noun compound bracketing. Such task consists in determining which nouns within a larger noun compound form subgroups. For example (from Lauer (1995)), *woman aid worker* would be bracketed as *woman [aid worker]*, called a right-bracketing, contrarily to *copper alloy rod*, which would be bracketed as *[copper alloy] rod*, called a left-bracketing.

In compound bracketing, when only three words are used, *n1 n2 n3*, the task becomes a binary decision between grouping *n1* and *n2* or grouping *n2* and *n3*. Two models, described in early work by Lauer (1995) and still used in recent work, are 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 attachment (modifying) noun relations, that of *n1 n3* and of *n2 n3*.

Most compound bracketing research has focus on three-noun compounds, often using a dataset from Lauer (1995). Some recent work (Pitler et al. (2010), Vadas and Curran (2007b)) looks at larger compounds, using a dataset created by Vadas and Curran (2007a) which we will also use in our research. For these larger noun compounds, for example *home market stock index futures trading* taken from the dataset, the adjacency model alone will not allow longer range dependencies to be taken into account. The bracketing algorithm we present mixes adjacency and dependency models, and looks at the complete expression to make its decisions. It relies on word pair association scores provided by different resources.

Among the resources used are web-corpus resources in the form of pre-processed ngrams. We look at Google ngrams and Google books ngrams (less often seen in use for different tasks). Then, a main contribution of this research is to introduce a linked-open data (LOD) resource and explore ways to use it to provide association scores.

Section 2 presents a short literature review, mostly from the perspective of resources used by different researchers for the task of compound bracketing. Section 3 presents the dataset used in our experiments. Section 4 describes the linked-open data and corpus-based resources we use. Section 5 defines association scores for each resource. Section 6 presents our bracketing method. Section 7 explains our evaluation approach and discusses the results obtained on the dataset. Section 8 concludes and discusses future work.

2. Related work

Noun bracketing has not receive as much attention as many other Natural Language Processing (NLP) task. Nakov and Hearst (2005) calls it an *understudied* language analysis problem. Nevertheless, a small body of work has emerged in the 1990s taking root and inspiration in earlier linguistic work (Levi, 1978). This body of work is expanding, using empirical methods which rely on the availability of larger and larger corpus.

Noun compound bracketing, sometimes referred to as NP parsing (Pitler et al., 2010), has been studied as a task in itself (e.g. Lauer (1995), Vadas and Curran (2007a), Nakov and Hearst (2005)). It is also studied as the first step of semantic analysis of NPs (Girju et al., 2005) where not only subgroups of words are found within the compound, but semantic relations between these groups are looked at (Nastase et al., 2013).

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 look at different resources and different methods for evaluating word-pair associations, since this is a core component in the problem's resolution steps.

Before the "Web-as-corpus" era, a first resource used in Lauer (1994) was the Grolier's encyclopedia. Processing of the resource found 35,974 noun sequences of which all but 655 were pairs. All pairs are considered non-ambiguous and could be used as observed data for the model. The au-

thor also used a second resource, the Rogets thesaurus[1], to provide association scores at a more conceptual level (instead of purely lexical). The Rogets thesaurus contains 1043 classes, with an average of 34 word nouns in each (from Lauer (1994)). This allows for some level of generalisation, as all nouns contained in a particular class contribute to the association scores of that class.

To make use of Rogets Thesaurus, Lauer's dataset was constrained to only pairs found in it, resulting in a list of 244 3-word noun compounds (216 unique). Lauer (1995) is one of the most cited author in noun bracketing research, and his gold standard has been used in different research articles (Lauer (1995),Lapata et al. (2004),Girju et al. (2005),Nakov and Hearst (2005)).

With the Web growing larger and being increasingly available, it has become a much used resource for providing noun pair association scores. Data sparseness is now less of an issue (at least for general language), and recent work tends to rely on lexical associations rather than depending on structured resources for generalisation. The work of Lapata et al. (2004) shows usefulness of web counts for different tasks, including compound noun bracketing. The work of Pitler et al. (2010) intensively uses web-scale ngrams in a supervised task for large NP bracketing, showing coverage impact on accuracy. Although coverage, even of web counts, will never be absolute, recent research tends to not constrain the dataset to the coverage of the resource.

Even with large coverage, large resources do not necessarily offer the same statistics, as they might have been constructed differently. For example, Vadas and Curran (2007a) use three large web-based resources: Google and MSN search engine hit counts and the Google Web 1T corpus, which contains n-gram counts collected from 1 trillion words of web text. Although very large, the correlation of bigram counts for a small dataset ((Lauer, 1994)) is high (over 0.90), but the correlation on their own larger dataset ranges between 0.60 (Google and Web1T) to 0.81 (MSN / Web 1T).

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 association estimates, and lead to better bracketing.

The use of web counts does not prevent the use of more structured resources. In (Vadas and Curran, 2007b), the use of web counts are combined with features from Wordnet (for a general language dataset) or to UMLS[2] (for a dataset of noun compounds extracted from biomedical texts).

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 differs from the original.

Length	Raw		Unique	
	Count	Ratio	Count	Ratio
3	4 114	62.3%	2889	60.95%
4	1 675	25.4%	1270	26.79%
5	547	8.3%	413	8.71%
6	225	3.4%	127	2.68%
7	36	<1%	32	<1%
8	6	<1%	5	<1%
9	4	<1%	4	<1%
Total	6600	100%	4749	100%

Table 1: Number of expressions of different size.

From this available file, we could construct the dataset for our experiments. In Vadas and Curran (2007a), some of the NP structures which they modified included determinants, numbers, punctuations or coordinations. We leave those out of our dataset to focus only on modified structures containing basic tags like common nouns (NN, NNS), proper nouns (NNP, NNPS), adverbs (RB, RBR) and adjectives (JJ, JJS). Two words expressions were removed as their bracketing is trivial. The construction method[3] of our dataset starts with the differential file published by Vadas and Curran (2007a) and extracts all expressions starting with the following tags: JJP, NML, NP-SBJ and NP. The expressions are then verified for completeness, so that the opening bracket should be closed within the length of text defined in the differential file. Groups which are not explicitly tagged (called implicit groups in (Vadas and Curran, 2007b)) are completed with the missing parentheses to produce the assumed right bracketing. For example, "(NML (NNP Nesbitt) (NNP Thomson) (NNP Deacon))" becomes "(NML (NNP Nesbitt) ((NNP Thomson) (NNP Deacon)))". Tags and single words enclosing parentheses are then removed from the extracted expression to produce a simplified version of the gold-standard bracketed expression including only the basic text and parentheses (i.e. "(Nesbitt (Thomson Deacon))").

The extraction produced a total 6,600 examples which we called the raw corpus. From these examples, we calculated duplicates expressions, which yielded a final test corpus of 4,749 unique expressions. Table 1 presents the number of examples in the datasets by length for the raw and unique corpora, and Table 2 gives examples for sizes 3 to 6. In Table 2, we purposely show common nouns and proper nouns to illustrate the existing variation within the current dataset. In later sections, we will discuss the coverage of resources and the relation between named entities and noun compound bracketing.

4. Resource description

In our present research, we investigate three resources for the compound bracketing task, focusing on their usefulness in an unsupervised approach. The first two are frequency-based of web-scale, namely the English Google

[1]The 1911 version is freely available online, at the ARTFL project, http://machaut.uchicago.edu/rogets

[2]The Unified Medical Language System (UMLS) is available at http://www.nlm.nih.gov/research/umls/

[3]Our method is published as part of the LREC resources sharing effort as a Java program to replicate our data extraction method. This will allow other researchers in the community to use the same data.

L	Example
3	(a) lung cancer deaths (b) Mary Washington College
4	(a) standardized achievement test scores (b) Fujitsu President Takuma Yamamoto
5	(a) annual gross domestic product growth (b) New York Stock Exchange issues
6	(a) general obligation distributable state aid bonds (b) Japanese auto maker Mazda Motor Corp

Table 2: Examples of expressions.

Web Ngrams ((Lin et al., 2010)), or GWN, and the English (non-fictional) Google Books ngrams ((Michel et al., 2011), or GBN. From this last resource, we compiled term frequencies by summing up all the years in which a term appears. This leads to a frequency-based resource referred to as GBN-A (all years). We also compiled term frequencies from the last 30 years, which we call GBN-R (for recent years).

The third resource is the open linked data DBpedia V3.9 which is based on the English Wikipedia pages. The main limitation in using structured resources is usually their lack of coverage. The knowledge acquisition bottleneck, referring to the gathering of large-scale coverage structured information, has often been cited as a major issue for NLP research. But in the past decade, collaborative world-wide efforts have allowed larger structured resources emerge, such as the Semantic Web. To the best of our knowledge, no previous research has used the semantic web for the noun bracketing task, and our research aims at introducing this resource and explore its usefulness.

4.1. Frequency-based resources

The GWN resource was generated from the set of archived pages used by the Google search engine in July 2009. They tokenized each page and summed up each word, number, punctuation and symbol and filtered any ngrams with a frequency count lesser than 200 occurrences. The GBN was created in the same way but the ngrams were generated from the archived electronic books from the Google Books website. As for the GWN, all character or number groups or single punctuation and symbol were added as a token. All case-aware ngrams were compiled separately and those that fell under the 40 mark were removed. Ngrams of length ranging from 1 to 5 were created for both of these resources.

For our purposes, both GWN and the GBN were filtered to remove entries which included numbers, parenthesis and symbols to be more manageable. As entries from both of these resources included multiple similar entries with different cases (i.e. test, TEST, Test), the interrogation technique was modified to add up all the frequencies from each similar entries. Table 3 shows the approximate count of expressions used in this research for the two frequency-based resource. GBN-A and GBN-R are represented as GBN in the table as they are both taken from this dataset.

Resource	1-gram count	2-gram count
GWN	>7.3 millions	>228.4 millions
GBN	>7.1 millions	>105.6 millions

Table 3: Number of one- and two-words expressions available in the frequency resources.

4.2. Linked open data resource

DBpedia[4] (Hellmann et al., 2009) is built from one of the largest resource on the web, Wikipedia. Many Wikipedia pages contain an Infobox (a small two-column table) to provide structured information about the entity described. All infoboxes are automatically parsed to generate DBpedia. DBpedia follows an RDF (Resource Description Format) representation, which is a W3C (World Wide Web Consortium) standard for the semantic web. DBpedia is growing every year, and the version we use (DBpedia V3.9) describes over 4 millions "'things"' for the English dataset, with many properties and links to other entities.

Components within the noun compounds to parse are found as entity names in DBpedia. For example *New York Medical School* contains two entity names *New_York* and *Medical_School* which exist in DBpedia. This split into components resembles query segmentation, useful in Information Retrieval. Query segmentation is "the process of taking a user's search engine query and dividing the tokens into individual phrases or semantic units" (Bergsma and Wang, 2007). Such segmentation reduces the complexity of the task, since in this example, the four word compound is reduced to two entity names *New_York Medical_School*, basically solving the bracketing problem.

But that would be too easy. Unfortunately, ambiguity comes into play. For example, *Mary Washington College*, leads to two competing interpretations, *Mary Washington_College* and *Mary_Washington College*. The complexity is therefore not that much reduced as we now work with competing entity names (rather than competing strings) which furthermore could each lead to multiple entities.

We differentiate entity names from entities. Entity names are surface forms that exists in DBpedia but they can lead to many different entities (word senses or actual named entities). There are usually disambiguation markers (e.g. New_York(disambiguation)) to show links between entity names and entities. There are also "redirects" links in DBpedia (and Wikipedia) which can be tricky to use as some of them are true synonyms (e.g. *automobile* and *car*) but others are just related items (e.g. *video* and *Audio-visual*).

Using a structured linked open data resource brings a completely new dimension, as we now work with entities and entity names instead of surface strings as for the frequency-based resources. Table 4 shows all existing entity names in DBpedia with their number of word senses for the complex compound *New York Stock Exchange Composite Trading*. Examples of the entities are also shown, to illustrate different relations between entity names and entities. Entity names can be considered abbreviations (*New - Net_economic_welfare*), shorter forms (*Exchange - Heat_exchange*), or domain specific terms (*Composite -*

[4]DBpedia is available at http://www.dbpedia.org

Resource	1-gram
GWN	92.70%
GBN	92.86%
DBpedia	88.78%

Table 5: Unigram coverage of our dataset.

Composite_(finance)). We also show some examples of redirects in Table 4 (indicated by "R").
The large number of entities (especially for single words) and the set of possible segmentations make the use of DBpedia for noun bracketing not at all trivial. Furthermore, we wish to make use of its structured data. In DBpedia, entities are linked via predicates, for example *Paris* would be linked to *France* using a predicate *capital-of*, or *located-in*. Such predicates provide paths between entities which we will use to measure their association scores. Our algorithm, presented in Section 5, will look into both segmentation and paths.

4.3. Resource coverage

Knowing the coverage of each resource provides an upper-bound on its usefulness to compute association measures. Table 5 shows resource coverage for the unigrams extracted from the noun compounds in our dataset. It is interesting to note that the GWN and GBN resources have very similar coverage for the unigrams, a marginal difference of only 0.16%. Datasets GBN-R and GBN-A both have the same coverage, represented as GBN in the table. DBpedia is not far behind, also providing a large coverage of unigrams. Table 6 presents a few examples to illustrate coverage differences between DBpedia and GWN/GBN. Here are a few types of coverage problems.

- Concatenations. For example *animal care*, and *department store*, found as written in one-word in GWN/GBN but not as an entity names in DBpedia.
- Tokenization. For example *U.S.A.* or *A.C.* will not be found as a unigram in GWN and GBN since the tokenization used in these resources includes punctuations.
- Plurals. Since we did not search for plurals in DBpedia, we lowered their coverage.
- Company names. Counter-intuitively, we assume DBpedia will contain companies, but maybe they are small and do not have their own entry, but they are "talked about" enough to be in GWN.
- People's names. Some links are not explicit in DBpedia (for *Biaggi* for example) even if a few people with last name *Biaggi* are in DBpedia.
- Part-of-speech. Some adjectives or adverbs (*extremely*, *interprovincial*, *award-winning*) will not be in DBpedia. Although, many adjective/adverbs are actually found because they also appear as nouns.

Some of these problems are relatively easy to look into in our future work, but others are not. The tokenization problem in GBN and GWN cannot really be resolved, since it is intrinsic to how the resources were built (e.g. U.S.A.). On the other hand, augmenting DBpedia coverage should be easily possible by searching for plurals, and for entities containing unigrams as part of their names.

Resource	Examples
DBpedia only	20th, A.C., B&H, black-and-white, U.S.A.
GWN/GBN only	agreements, animalcare, Biaggi, departmentstore, extremely, Intelogic, Interprovincial
neither	achievement-test, award-winning,

Table 6: Examples of unigrams missing in different resources.

Size	Examples
3	magnetic resonance imaging nuclear power plant Vincent van Gogh International Monetary Fund
4	The Wall Street Journal New Jersey Turnpike Authority Carlos Salinas de Gortari Ho Chi Minh City
5	Defense Advanced Research Projects Agency Pennsylvania State Employees Retirement System real estate mortgage investment conduit New York State Supreme Court
6	Ateliers de Constructions Mecaniques de Vevey St. Johns River Water Management District

Table 7: Examples of larger entities found in DBpedia.

4.4. Named entities

Many named entities are found in the Penn Treebank (PTB), like cities, people names, or company names. A manual analysis of our subset of PTB showed that 5,286 out of 6,600 expressions (80.09%) contained at least one named entity. While it is not surprising for texts in the news domain, this proportion of named entities is not representative of texts found in other domains.
For named entities, we would expect an entity-oriented resource such as DBpedia to be useful. Many composite named entities like *Los Angeles Mayor Tom Bradley* (annotated as a noun compound in the revised PTB) are not found as complete expression but *Los Angeles* and *Tom Bradley* are found separately. DBpedia does contain entities of larger sizes, as shown in Table 7. In our bracketing approach, since we try to find dependencies between all word-pairs, we do not use entity names with more than two words.
This issue about named entities and noun compound bracketing is complex. It is discussed a bit in (Vadas and Curran, 2007a), as they used a NE annotator to suggest bracketing to the annotators (who could accept or reject them). The entity types used were the ones defined in (Weischedel and Ada Brunstein, 2005) (e.g. Person, Facility, Organization, Nationality, Product, Event, etc). Named entities could be kept "as is" by the annotators. In our dataset, we trans-

Surface found	Nb Senses found	Nb Redirects	Examples
New	14	6	New_(surname), Net_economic_welfare
York	88	1	Yorktown, University_of_York
Stock	87	0	Livestock, Stock_(album)
Exchange	16	1	Heat_exchange, Exchange_(chess)
Composite	15	0	Composite_material, Composite_(finance)
Trading	0	1	(R)Trade
New York	29	1	New_York_City, New_York_(U2_song)
Stock Exchange	0	1	(R)Stock_exchange
Stock Trading	0	1	(R)Stock_trader
Exchange Trading	0	1	(R)Stock_exchange
New York Stock Exchange	0	1	(R)New_York_Stock_Exchange

Table 4: All entities found for *New York Stock Exchange Composite Trading*.

Measure	Formulae
Chi square	$\frac{N*(O_{11}O_{22}-O_{12}O_{21})^2}{(O_{11}+O_{12})(O_{11}+O_{21})(O_{12}+O_{21})(O_{21}+O_{22})}$
PMI	$log_2(\frac{P(w_1,w_2)}{P(w_1)*P(w_2)})$
Dice	$\frac{2*Bi(w_1,w_2)}{Uni(w_1)+Uni(w_2)}$

Table 8: Association measure formulae used in the study.

formed those into right bracketed, as we wanted to have all expressions fully bracketed. This will have an impact on our results, and we will revisit this decision in future work, as we study more closely this relation between named entities and compositionality of noun compounds.

5. Association measures

A first step for noun bracketing, as we emphasized in Section 2, is to establish association scores between nouns using different resources and measures. Since we use both web corpus data (unstructured), and a link-open data (structured), we present two different ways of calculating association scores. We also discuss the fact that in DBpedia, we must deal with entities described and not surface forms.

5.1. Frequency-based resources

For our two frequency-based resources, we calculate association strength using the Chi square (used in (Vadas and Curran, 2007b)), the pointwise mutual information (PMI) (used both in Nakov and Hearst (2005) and Pitler et al. (2010)), as well as the Dice measure. These statistical measures were used on both the GWN and the GBN (-A and -R) resources. These three measures are defined in Table 8. The Chi square measure refers to a 2x2 table of bigram occurrences for the four frequencies of bigrams containing both words (O_{11}), none of the two words (O_{22}), the first word but the the second (O_{21}) or the second but not the first (O_{12}). In this formulae, N refers to the total number of bigrams in the resource and O_{nm} refers to the frequency count found in the 2x2 table at the N^{th} row and the M^{th} column. The PMI measure applies a binary-based log to the bigram probability divided by the product of its unigrams probabilities. The Dice measure uses twice the raw frequency of the studied bigram divided by the sum of the frequency of its unigrams.

5.2. Linked open data resource

As mentioned in Section 4.2, calculating association scores using DBpedia can take on many forms. For the present exploration, we construct our algorithm to combine two hypothesis. The first one is to minimize the number of entity names found in the expression. The second one is to maximize the number of *valid paths* among the entities represented by the entity names. As we saw earlier, many entity names (either single or multi-words) are ambiguous and refer to different possible entities (such as in Table 4).

Our definition of a valid path is limited to two possibilities. First, both entities are part of the same triple. Second, both entities are part of different triples sharing a subject or an object. For example, given the two triples (New_York, located_in, United_States) and (Chicago, located_in, United_States), there is a valid path between New_York and United_States (same triple) and also a valid path between New_York and Chicago (shared object). In the present work, the two types of paths are counted equally, but future work could assign different weights to them.

Let us illustrate with an example, in which we will use MaxNbPaths(X,Y) to refer to the number of valid paths between entity names X and Y. Given compound expression "ABCD", let us assume all unigrams "A", "B", "C" and "D" exist as entity names. Let us also assume bigrams "AB" and "CD" also exist as entity names. Then three segmentations (S1, S2, S3) are possible: S1(AB,C,D), S2(A,B,CD) or S3(A,B,C,D). The first two segmentations (S1 and S2) minimize the number of entity names and will be kept for path calculation. For each of S1 and S2, the association strength between each pair of entity names will be given by the maximum number of paths among any two of their entities. In Figure 1, we illustrate a possible case for S1, assuming AB links to a single entity, C links to 3 entities, and D links to 2 entities. We calculate MaxNbPaths(AB,C) + MaxNbPaths(AB,D) + MaxNbPaths(C,D) to obtain a score of S1. The same will be performed for segmentation S2, and we keep the segmentation with the highest score. This best segmentation moves to the second step of actual bracketing, explained in Section 6, providing its MaxNbPaths as association scores to the bracketing algorithm.

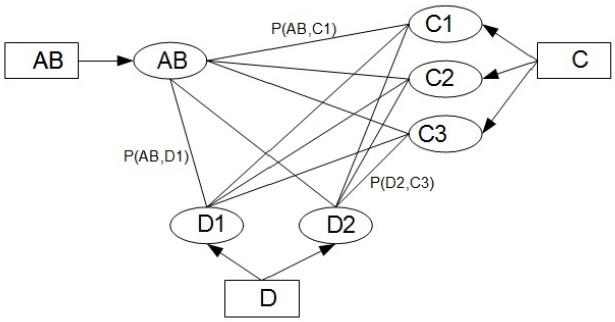

Figure 1: Example of path strength evaluation in DBpedia

Resource	Algorithm	Strict	Lenient
Baseline	Right	13.74%	24.12%
	Left	52.06%	66.23%
DBpedia	Path	54.08%	64.60%
GBN-A	chi	60.18%	72.33%
	pmi	61.04%	73.20%
	dice	59.87%	72.11%
GBN-R	chi	60.14%	72.26%
	pmi	61.04%	73.13%
	dice	59.82%	72.17%
GWN	chi	54.43%	66.63%
	pmi	60.41%	72.47%
	dice	51.80%	63.90%

Table 9: Comparing strict and lenient evaluation results.

6. Bracketing method

As in the work of Pitler et al. (2010), our bracketing algorithm looks at the whole expression for its evaluation. This is different form the algorithm suggested in (Barker, 1998) and used in (Vadas and Curran, 2007b) which only uses local information (three-words at a time, in a right-to-left moving window).

Our algorithm consists in creating a list (L1) containing every word pairs that can be generated from an expression. L1 is then sorted in decreasing order of association scores. The score of each pair is provided either from GBN-A, GBN-R, GWN or Dbpedia and is calculated using one of the methods detailed in Section 5. In our algorithm, association scores are considered as dependency scores, that is modifier/head scores. For example, an expression "1 2 3 4 5" would generate a list L1 containing {(1,2), (1,3), (1,4), (1,5), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5)} and each possible modifier/head pair would be scored using a specific association measure and ordered into L1 in a descending order.

From there, we construct the final list of dependencies (L2), which will define the complete bracketing of the expression. This is done by selecting in order each word pair (A,B) from L1 and adding 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 (e.g. if L2 already contains (12)(3(45)), then (24) would create an invalid crossing and is not accepted). The selection of pairs from L1 is ended when all words are used as modifiers in L2, except for the right-most one in the expression.

Our algorithm is greedy as only the best score (enabling a valid integration of the word pair into L2) is chosen at every step without consideration for the actual distance between the two words in the source expression. This helps linking far reaching dependencies in noun compounds, but it might also force some strong association between two distant words without regard to the soundness of using nearer words.

7. Evaluation

Two methods are used to evaluate the bracketing algorithm against the gold standard. The first method is a strict match, like the exact evaluation method used in Vadas and Curran (2007b). It requires a complete and exact match for all the groups found in the reference expression without considering the tags. The final score is thus the number of correctly bracketed expressions divided by the number of inspected expressions.

The second method, called lenient, checks for each parenthesis group of an expression and compares it to the gold standard. For example, a six word long candidate "'((((A B) C) D) E) F"' (groups: [AB], [ABC], [ABCD], [ABCDE]) compared to a gold standard "'(((A B)(C D)) E) F"' (groups: [AB], [CD], [ABCD], [ABCDE]) would score a recall of 3/4 = 75% as three groups are correct compared to the four in the gold. As both the gold and test expressions (of length N) are fully bracketed, the number of groups (N-1 excluding the top level group) are always the same in both expressions and thus, precision and recall will be the same as the F-measure.

Our suggested lenient evaluation is different, and more severe, than looking at word relations in a binary tree. Using the previous example, the gold expression would give A-B, B-D, C-D, D-E and E-F and the test expression would give A-B, B-C, C-D, D-E, E-F which would give a score of 80%. Furthermore, compared to a gold three word expression ((A B) C), a test bracketing of (A (B C)) would obtain 0% for the lenient, as the test expression would miss the only non-trivial choice to be made, but the binary tree evaluation method would give 50% as B-C is still present in both cases.

We first show, in Table 9 the comparative results from the three resources, for a strict or lenient evaluation. Two baselines were also calculated, with a default right and left bracketing. Following the findings by Lauer (1995), the left-bracketing is much more common in our dataset. Compared to the left-bracketing baseline, all methods score a bit over for the strict evaluation, and all but DBpedia score again over in the lenient evaluation.

The closest research providing comparable results on longer compounds are Vadas and Curran (2007b) and Pitler et al. (2010), although both focus on supervised approaches, and furthermore, Vadas and Curran (2007b) uses contextual features, assuming the noun compounds are to be bracketed in context. Still, we can compare to the results for unsupervised given in Vadas and Curran (2007b). They report exact match for complex NPs to be 54.66 for Base-

L	Rand.	BL	DBpedia	GWN pmi	GBN-A pmi	GBN-R pmi
3	50%	79.2%	69.86%	80.68%	81.23%	81.47%
4	20%	12.67%	37.67%	36.76%	37.70%	37.23%
5	7.1%	0.73%	15.46%	13.53%	13.77%	13.53%
6	2.4%	0.0%	0.75%	6.06%	6.06%	6.06%
7	0.8%	0.0%	3.13%	3.13%	6.25%	6.25%
8	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%
9	<0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
All	36.88%	52.06	54.08%	60.41%	61.04%	61.04%

Table 10: Strict evaluation results, per expression size.

line (right branching), 32.66 chi-square dependency and 35.86 chi-square adjacency. As we obtain around 60% for strict matches, we seem well-above the unsupervised approach they used, which combines their association scores using an implementation of Barker's algorithm.

As mentioned in Pitler et al. (2010), the number of possible binary trees (possible bracketing) increases with the Catalan numbers[5] meaning random results as shown in the second column (named "'Rand."') of Table 10. Results by noun compound length is shown for the left bracketing (BL), DBpedia, as well as the best measure for GWN (pmi) and GBN (chi square). Results were good for the baseline on 3 words expressions but degraded very quickly for longer expressions. All methods did better than random and baseline of lengths from 4 to 7 for the strict evaluation.

The top row of Table 10, shows results above 80% obtained (for GBN and GWN) on the 3-word compounds. This is comparable to results in Vadas and Curran (2007b) of around 80% with voters (dependency and adjacency). DBpedia does not perform as well on 3-word compounds, but does on the larger ones, probably showing the usefulness of detecting entity names within the expression.

To get a better sense of the results achieved using each resource, we show in Table 11 the bracketed outputs for the examples previously given in Table 2.

8. Discussion and Conclusion

Even if bracketing of three-word expressions have been performed quite successfully using unsupervised approaches using 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 exploring other resources, but keeping an unsupervised approach, to make our method independent of any dataset.

Our research provides a first exploration of the usefulness of DBpedia for the noun bracketing task. Although providing lower results on three-word expressions, DBpedia does provide reasonable results on larger expressions, even though entity names larger than two words have not even been used in our experiment.

We measured that out of 6600 queries, DBPedia found at least one entity name of two words in 65% of them. It found sometimes 2 entity names, for a total of 5800. We have started a discussion on the relation between named entities and bracketing issues, but we hope to further investigate this issue, and the related issue of determining compounds on which DBPedia does well compared to GWN/GBN (and vice-versa). DBpedia, built from Wikipedia, has grown large enough to allow coverage near the one of GWN/GBN. Eventually, we believe an hybrid model, built after a good understanding of the strength and weaknesses of each resource, will provide a good solution to the noun compound bracketing problem. Within that hybrid model, individual models should also take further advantage of the individual resources. For the frequency-based resource, different searches (as suggested in Vadas and Curran (2007b)) such as simple paraphrases, could be tested. For DBpedia, our simple valid path count algorithm should be revisited to make better use of different path lengths and path types.

[5] see http://en.wikipedia.org/wiki/Catalan_number

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.*

10. References

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

Bergsma, S. and Wang, Q. (2007). Learning Noun Phrase Query Segmentation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, number June, pages 819–826.

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

Hellmann, S., Stadler, C., and Lehmann, J. (2009). DBpedia Live Extraction. *On the Move to Meaningful Internet Systems OTM 2009*, 5871(0):1209–1223.

Lapata, M., St, P., Sheffield, S., and Keller, F. (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.

L	Resource	Example
3	Gold	(lung cancer) deaths
	GWN	(lung cancer) deaths
	GBN	(lung cancer) deaths
	DBpedia	(lung cancer) deaths
	Gold	(Mary Washington) College
	GWN	(Mary Washington) College
	GBN	Mary (Washington College)
	DBpedia	(Mary Washington) College
4	Gold	((standardized (achievement test)) scores)
	GWN	((standardized (achievement test)) scores)
	GBN	(((standardized achievement) test) scores)
	DBpedia	(standardized (achievement (test scores)))
	Gold	((Fujitsu President) (Takuma Yamamoto))
	GWN	((Fujitsu President) (Takuma Yamamoto))
	GBN	((Fujitsu President) (Takuma Yamamoto))
	DBpedia	(Fujitsu (President (Takuma Yamamoto)))
5	Gold	(annual ((gross (domestic product)) growth))
	GWN	(annual (((gross domestic) product) growth)))
	GBN	(annual (((gross domestic) product) growth))
	DBpedia	((annual gross)((domestic product) Growth))
	Gold	(((New York) (Stock Exchange)) issues)
	GWN	((((New York) Stock) Exchange) issues)
	GBN	((((New York) Stock) Exchange) issues)
	DBpedia	(((New York) (Stock Exchange))issues)
6	Gold	((general obligation) ((distributable ((state aid) bonds))))
	GWN	((general obligation) (((distributable state) aid) bonds))
	GBN	((general obligation) (((distributable state) aid) bonds))
	DBpedia	(((General Obligation) Distributable)((State Aid) Bonds))
	Gold	((Japanese (auto maker)) ((Mazda Motor) Corp))
	GWN	(((((Japanese auto) maker) Mazda) Motor) Corp)
	GBN	(((((Japanese auto) maker) Mazda) Motor) Corp)
	DBpedia	(Japanese (((auto maker) (Mazda Motor))Corp))

Table 11: Examples of bracketed expressions using different resources

Lauer, M. (1994). Conceptual association for compound noun analysis. *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 337–339.

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

Levi, J. (1978). *The syntax and semantics of complex nominals*.

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

Marcus, M. P., Beatrice, S., and Marcinkiewicz, M. A. (1994). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

Michel, J., Shen, Y., Aiden, A., and Veres, A. (2011). Quantitative analysis of culture using millions of digitized books. *science*.

Nakov, P. and Hearst, M. (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.

Nastase, V., Nakov, P., O Seaghdha, D., and Szpakowicz, S. (2013). *Semantic Relations Between Nominals*. Morgan and Claypool Publishers.

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

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

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

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