UCD-Goggle: A Hybrid System for Noun Compound Paraphrasing

Guofu Li
School of Computer Science
and Informatics
University College Dublin
guofu.li@ucd.ie

Alejandra Lopez-Fernandez
School of Computer Science
and Informatics
University College Dublin
alejandra.lopez
-fernandez@ucd.ie

Tony Veale
School of Computer Science
and Informatics
University College Dublin
tony.veale@ucd.ie

Abstract

This paper addresses the problem of ranking a list of paraphrases associated with a noun-noun compound as closely as possible to human raters (Butnariu et al., 2010). UCD-Goggle tackles this task using semantic knowledge learnt from the Google n-grams together with human-preferences for paraphrases mined from training data. Empirical evaluation shows that UCD-Goggle achieves 0.432 Spearman correlation with human judgments.

1 Introduction

Noun compounds (NC) are sequences of nouns acting as a single noun (Downing, 1977). Research on noun compounds involves two main tasks: NC detection and NC interpretation. The latter has been studied in the context of many natural language applications, including question-answering, machine translation, information retrieval, and information extraction.

The use of multiple *paraphrases* as a semantic intepretation of noun compounds has recently become popular (Kim and Baldwin, 2006; Nakov and Hearst, 2006; Butnariu and Veale, 2008; Nakov, 2008). The best paraphrases are those which most aptly characterize the relationship between the *modifier* noun and the *head* noun.

The aim of this current work is to provide a ranking for a list of paraphrases that best approximates human rankings for the same paraphrases. We have created a system called UCD-Goggle, which uses semantic knowledge acquired from Google n-grams together with human-preferences mined from training data. Three major components are involved in our system: B-score, produced by a Bayesian algorithm using semantic knowledge from the n-grams corpus with a smoothing layer of additional inference; R_t-score captures human preferences observed in the tail distribution of training data; and R_p-score captures pairwise paraphrase preferences calculated from the training data. Our best system for SemEval-2 task 9 combines all three components and achieves a Spearman correlation of 0.432 with human rankings.

This paper is organized as follows: the Bayesian B-score is introduced in section 2. In section 3 we describe two supervised approaches to mining the preferences of human raters from training data. Finally, section 4 presents the results of our empirical evaluation of the UCD-Goggle system.

2 Semantic Approach

2.1 Collecting Data

Google have made their web n-grams, also known as Web-1T corpus, public via the Linguistic Data Consortium (Brants and Franz, 2006). This corpus contains sequences of n terms that occur more than 40 times on the web.

We view the paraphrase task as that of suggesting the right verb phrase for two nouns (Butnariu and Veale, 2008). Previous work has shown the n-grams corpus to be a promising resource for retrieving semantic evidence for this approach. However, the corpus itself needs to be tailored to serve our purpose. Since the n-grams corpus is a collection of raw snippets from the web, together with their web frequency, certain pre-processing steps are essential before it can be used as a semi-structured knowledge base. Following a syntactic pattern approach, snippets in the n-grams that agree with the following patterns are harvested:

1. *Head VP Mod*
2. *Head VP DET Mod*
3. *Head* $[that|which]$ *VP Mod*
4. *Head* $[that|which]$ *VP DET Mod*

Here, *DET* denotes any of the determiners (*i.e.*,

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 230–233,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

the set of $\{an, a, the\}$ for English), *Head* and *Mod* are nouns for heads and modifiers, and *VP* stands for verb-based paraphrases observed in the test data. It must be highlighted that, when we collect snippets for the KB, any *Head* or *Mod* that falls out of the range of the dataset are also accepted via a process of semantic slippage (to be discussed in Sect. 2.4). The patterns listed above enable us to collect examples such as:

1. *"bread containing nut"*
2. *"pill alleviates the headache"*
3. *"novel which is about crimes"*
4. *"problem that involves the students"*

After a shallow parse, these snippets are formalized into the triple format $\langle Head, Para, Mod\rangle$. The sample snippets above are represented as:

1. $\langle$`bread, contain, nut`$\rangle$
2. $\langle$`pill, alleviate, headache`$\rangle$
3. $\langle$`novel, be about, crime`$\rangle$
4. $\langle$`problem, involve, student`$\rangle$

We use $\|Head, Para, Mod\|$ to denote the frequency of $\langle Head, Para, Mod\rangle$ in the n-grams.

2.2 Loosely Coupled Compound Analysis

Tens of millions of snippets are harvested and cleaned up in this way, yet expecting even this large set to provide decent coverage over the test data is still unrealistic. We calculated the probability of an example in the test data to appear in KB at less than 1%. To overcome the coverage issue, a loosely coupled analysis and representation of compounds is employed. Despite the fact that both modifier and head can influence the ranking of a paraphrase, we believe that either the modifier or the head is the dominating factor in most cases. This assumption has been shown to be plausible by earlier work (Butnariu and Veale, 2008). Thus, instead of storing complete triples in our KB, we divide each complete triple into two partial triples as shown below:

$$\langle Head, Para, Mod\rangle \rightarrow \begin{cases} \langle Head, Para, ?\rangle \\ \langle ?, Para, Mod\rangle \end{cases}$$

We can also retrieve these partial triples directly from the n-grams corpus using partial patterns like *"Head Para"* and *"Para Mod"*. However, just as shorter incomplete patterns can produce a larger KB, they also accept much more noise. For instance, single-verb paraphrases are very common among the test data. In these cases, the partial pattern approach would need to harvest snippets with the form *"NN VV"* or *"VV NN"* from 2-grams, which are too common to be reliable.

2.3 Probabilistic Framework

In the probabilistic framework, we define the B-score as the conditional probability of a paraphrase, $Para$, being suggested for a given compound $Comp$:

$$B(Para; Comp) \equiv P(Para|Comp) \quad (1)$$

Using the KB, we can estimate this conditional probability by applying the Bayes theorem:

$$P(Para|Comp) = \frac{P(Comp|Para)P(Para)}{P(Comp)} \quad (2)$$

The loose-coupling assumption (Sect. 2.2) allows us to estimate $P(Comp)$ as:

$$P(Comp) \equiv P(Mod \vee Head). \quad (3)$$

Meanwhile, *a priori* probabilities such as $P(Para)$ can be easily inferred from the KB.

2.4 Inferential Smoothing Layer

After applying the loose-coupling technique described in Section 2.2, the coverage of the KB rises to 31.78% (see Figure 1). To further increase this coverage, an inference layer is added to the system. This layer aims to stretch the contents of the KB via semantic slippage to the KB, as guided by the maximization of a fitness function. A WordNet-based similarity matrix is employed (Seco et al., 2004) to provide a similarity measure between nouns (so $sim(x, x)$ is 1). Then, a superset of *Head* or *Mod* (denoted as $\mathcal{H}$ and $\mathcal{M}$ respectively) can be extracted by including all nouns with similarity greater than 0 to any of them in the test data. Formally, for *Head* we have:

$$\mathcal{H} = \{h|sim(h, Head) \geq 0, Head \text{ in dataset}\}. \quad (4)$$

The definition of $\mathcal{M}$ is analogous to that of $\mathcal{H}$.

A system of equations is defined to produce alternatives for *Head* and *Mod* and their smoothed corpus frequencies (we show only the functions for head here):

$$h_0 = Head \quad (5)$$

$$fit(h) = sim^2(h, h_n) \times \|h, p, ?\| \quad (6)$$

$$h_{n+1} = \arg\max_{h \in \mathcal{H}} fit(h) \quad (7)$$

Here, $fit(h)$ is a fitness function of the candidate head h, in the context of a paraphrase p. Empirically, we use h_1 for $Head$ and $fit(h_1)$ for $\|Head, Para, ?\|$ when calculating the B-score back in the probabilistic framework (Sect. 2.3). In theory, we can apply this smoothing step repeatedly until convergence is obtained.

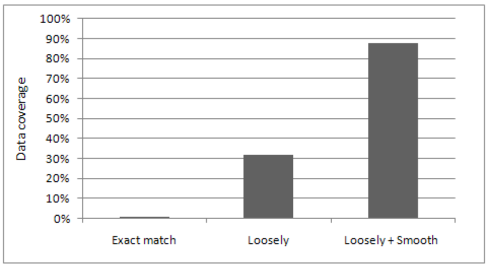

Figure 1: Comparison on coverage.

This semantic slippage mechanism allows a computer to infer the missing parts of the KB, by building a bridge between the limitations of a finite KB and the knowledge demands of an application. Figure 1 above shows how the coverage of the system increases when using partial matching and the smoothing technique, over the use of exact matching with the KB.

3 Preferences for Paraphrases

3.1 Tail-based Preference

Similar to various types of data studied by social scientists, the distribution of strings in our corpus tends to obey Zipf's law (Zipf, 1936). The same Zipfian trend was also observed in the compound-paraphrase dataset: more than 190 out of 250 compounds in the training data have 60% of their paraphrases in an undiscriminating tail, while 245 of 250 have 50% of their paraphrases in the tail. We thus assume the existence of a long *tail* in the paraphrase list for each compound.

The tail of each paraphrase list can be a valuable heuristic for modeling human paraphrase preferences. We refer to this model as the *tail-based preference* model. We assume that an occurrence of a paraphrase is deemed to occur in the tail *iff* it is mentioned by the human raters only once. Thus, the tail preference is defined as the probability that a paraphrase appears in the non-tail part of the list for all compounds in the training data. Formally, it can be expressed as:

$$R_t(p) = \frac{\sum\limits_{c \in \mathcal{C}} \delta(c,p) f(c,p)}{\sum\limits_{c \in \mathcal{C}} f(c,p)} \tag{8}$$

where $\mathcal{C}$ is the set of all compounds in the training data and $f(c,p)$ is the frequency of paraphrase p on compound c as given by the human raters. The $\delta(c,p)$ is a filter coefficient as shown below:

$$\delta(c,p) = \begin{cases} 1, & f(c,p) > 1, \\ 0, & f(c,p) = 1. \end{cases} \tag{9}$$

The *tail-based preference* model is simple but effective when used in conjunction with semantic ranking via the KB acquired from n-grams. However, an important drawback is that the tail model assigns a static preference to paraphrase (i.e., tail preferences are assumed to be context-independent). More than that, this preference does not take information from non-tail paraphrases into consideration. Due to these downsides, we use pairwise preferences described below.

3.2 Pairwise Preference

To fully utilize the training data, we employ another preference mining approach called *pairwise preference* modeling. This approach applies the principle of *pairwise comparison* (David, 1988) to determine the rank of a paraphrase inside a list.

We build a pairwise comparison matrix Π for paraphrases using the values of Equation 10 (here we have assumed that each of the paraphrases has been mapped into numeric values):

$$\Pi_{i,j} = \begin{cases} \frac{n(p_i,p_j)}{n(p_i,p_j)+n(p_j,p_i)}, & n(p_i,p_j) > n(p_j,p_i), \\ 0, & otherwise. \end{cases} \tag{10}$$

where $n(p_i, p_j)$ is the relative preferability of p_i to p_j. To illustrate the logic behind $n(x, y)$, we imagine a scenario with three compounds shown in Table 1:

	abor. prob.	*abor. vote*	*arti. desc.*
involve	12	8	3
concern	10	9	5
be about	3	9	15

Table 1: An example[1] to illustrate $n(x, y)$

[1] In this example, *abor. prob.* stands for *abortion problem*, *abor. vote* stands for *abortion vote*, and *arti. desc.* stands for *artifact description*

The relative preferability is given by the number of times that the frequency of p_i from human raters is greater than that of p_j. Observing that 1 out of 3 times *involve* is ranked higher than *concern*, we can calculate their relative preferability as:

$$\begin{aligned} n(involve, concern) = & \; 1 \\ n(concern, involve) = & \; 2 \end{aligned}$$

Once the matrix is built, the preference score for a paraphrase i is calculated as:

$$R_p(i; c) = \frac{\sum_{j \in \mathcal{P}_c} \Pi_{i,j}}{|\mathcal{P}_c|} \tag{11}$$

where $\mathcal{P}_c$ is the list of paraphrases for a given compound c in the test data. The pairwise preference puts a paraphrase in the context of its company, so that the opinions of human raters can be approximated more precisely.

4 Empirical Results

We evaluated our system by tackling theSemEval-2 task 9 test data. We created three systems with different combinations of the three components (B, R_t, R_p). Table 2 below shows the performance of UCD-Goggle for each setting:

	System Config	Spearman ρ	Pearson r
I	$B + R_t$	0.380	0.252
II	R_p	0.418	0.375
III	$B + R_t + R_p$	**0.432**	**0.395**
*	Baseline	0.425	0.344

Table 2: Evaluation results on different settings of the UCD-Goggle system.

The first setting is a hybrid system which first calculates a ranking according to the $n$grams corpus and then applies a very simple preference heuristic (Sect. 2.3 and 3.1). The second setting simply applies the pairwise preference algorithm to the training data to learn ranking preferences (Sect. 3.2). Finally, the third setting integrates both of these settings in a single approach.

The individual contribution of B-score and R_t was tested by two-fold cross validation applied to the training data. The training data was split into two subsets and preferences were learnt from one part and then applied to the other. As an unsupervised algorithm, B-score produced Spearman correlation of 0.31 while the R_t-score gave 0.33. We noticed that more than 78% of the paraphrases had 0 score by R_t. This number not only reconfirmed the existence of the long-tail phenomenon, but also suggested that R_t-score alone could hardly capture the preference on the non-tail part. On the other hand, with more than 80% chance we could expect B to produce a non-zero score for a paraphrase, even if the paraphrase fell out of the topic. When combined together, B and R_t complemented each other and improved the performance considerably. However, this combined effort still could not beat the pairwise preference R_p or the baseline system, which had no semantic knowledge involved. The major limitation of our system is that the semantic approach is totally ignorant of the training data. In future work, we will intend to use it as a valuable resource in both KB construction and ranking stage.

References

T. Brants and A. Franz. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium.

C. Butnariu and T. Veale. 2008. A concept-centered approach to noun-compound interpretation. In *Proc. of the 22nd COLING*, pages 81–88, Manchester, UK.

C. Butnariu, S. N. Kim, P. Nakov, D. Ó Séaghdha, S. Szpakowicz, and T. Veale. 2010. Semeval-2 task 9: The interpretation of noun compounds using paraphrasing verbs and prepositions. In *Workshop on Semantic Evaluation*, Uppsala, Sweden.

H. A. David. 1988. *The Method of Paired Comparisons*. Oxford University Press, New York.

P. Downing. 1977. On the creation and use of English compound nouns. In *Language 53*, pages 810–842.

S. N. Kim and T. Baldwin. 2006. Interpreting semantic relations in noun compounds via verb semantics. In *Proc. of the COLING/ACL*, pages 491–498, Morristown, NJ, USA.

P. Nakov and M. A. Hearst. 2006. Using verbs to characterize noun-noun relations. In *Proc. of AIMSA*, pages 233–244.

P. Nakov. 2008. Noun compound interpretation using paraphrasing verbs: Feasibility study. In *Proc. of the 13th AIMSA*, pages 103–117, Berlin, Heidelberg. Springer-Verlag.

N. Seco, T. Veale, and J. Hayes. 2004. An intrinsic information content metric for semantic similarity in WordNet. In *Proc. of the 16th ECAI*, Valencia, Spain. John Wiley.

G. K. Zipf. 1936. *The Psycho-Biology of Language: An Introdution to Dynamic Philology*. Routledge, London.