

# Pattern-Based Synonym and Antonym Extraction

Wenbo Wang  
Kno.e.sis Center  
Wright State University  
Dayton, OH, USA  
wenbo@knoesis.org

Christopher Thomas  
Kno.e.sis Center  
Wright State University  
Dayton, OH, USA  
topher@knoesis.org

Amit Sheth  
Kno.e.sis Center  
Wright State University  
Dayton, OH, USA  
amit@knoesis.org

Victor Chan  
AFRL Wright Patterson AFB  
Dayton, OH, USA  
victor.chan@wpafb.af.mil

## ABSTRACT

Many research studies adopt manually selected patterns for semantic relation extraction. However, manually identifying and discovering patterns is time consuming and it is difficult to discover all potential candidates. Instead, we propose an automatic pattern construction approach to extract verb synonyms and antonyms from English newspapers. Instead of relying on a single pattern, we combine results indicated by multiple patterns to maximize the recall.

## Categories and Subject Descriptors

I.2.6 [Learning]: Knowledge acquisition; H.3.1 [Content Analysis and Indexing]: Thesauruses—*verb synonyms, verb antonyms*

## General Terms

Algorithms, Design, Languages

## Keywords

Information Extraction, Verb Relationship, Pattern-Based extraction

## 1. INTRODUCTION

Verbs play a vital role in natural language, as they express actions, events, and states. Verbs, just like words from other linguistic categories, can be related through synonymy, antonymy or other relations. To facilitate many NLP, IR, or IE tasks, such as question answering system, machine translation, it is useful to know the underlying relationships between verbs. This is particularly important, because depending on the domain, some inter-verb relationships may be more or less common. Verbs that are synonymous in a general context may indicate subtle, but important differences in a specific context, for example in the biomedical

domain [8]. For this reason, it is important to not only rely on general dictionaries, such as WordNet [3], but to analyze text corpora that are domain-specific. In this paper, we focus on verb synonymy/antonymy (SYN/ANT) relationship extraction.

The problem of inter-verb relationship learning can be considered a type of information extraction problem (such as location name extraction [2] and hyponym extraction [7]). This means that existing approaches such as pattern based extraction, can be applied to our problem. On the other hand, our problem has some specific challenges. For example the frequency at which people use two or more synonymous verbs in one sentence is much less than the frequency at which people use related nouns or antonym verbs. We often see sentences such as “*fruits such as strawberries, apples, and oranges*”. Antonym verbs are also used frequently because they can highlight opposite ideas, such as “*either live or die*”. However, synonymous verbs are used less often in the same sentence because people use verbs to communicate different ideas, while the function of synonym verbs is to emphasize the same idea in different ways.

The rest of this paper is structured as follows: Section 2 discusses related work. We will explain how we automatically extract verb relationships in Section 3. In Section 4, we discuss the achievements and show possible future research directions.

## 2. RELATED WORK

Pattern based information extraction has been described extensively in the research literature. Hearst [7] applied patterns to extract hyponymy relationship, using patterns such as “ $NP_1$ , such as  $NP_2$ ” to infer that  $NP_2$  indicates hyponym(s) of  $NP_1$ , where  $NP$  indicates a noun phrase. From the sample text “*Students, such as sophomores*”, we can infer that a sophomore is a type of student. Chklovski et al. [1] apply the same approach to inter-verb-relationship extraction on a Web-scale corpus. They first collect highly associated verb pairs as candidates and then formulate Web search queries by instantiating predefined patterns with verb pairs. The more frequently a verb pair co-occurs with a pattern, the more probable it will have the relationship indicated by the pattern. Both approaches rely on patterns discovered with human efforts, and there is no guarantee that those patterns are comprehensive and complete.

Other research ventured into learning patterns from known

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACMSE '10 April 15-17, 2010, Oxford, MS, USA.

Copyright ©2010 ACM 978-1-4503-0064-3/10/04 ...\$10.00.

facts. Ravichandran et al.[10] use a confidence score to specify how good a learned pattern is. A pattern with the highest score out of all the patterns associated with the verb pair is used to indicate a relationship. The study tackled the problem of how reliable a single pattern is but leaves the question open of how to accumulate confidence when a term pair is found with several patterns.

Other approaches to information extraction use contextual information of single terms. The *distributional hypothesis* [6] states that similar words usually share similar contexts. Hagiwara[5] measures the context similarity by *distributional features*, and utilizes a Support Vector Machine (SVM) to separate synonym pairs from other pairs. Mirkin et al.[9] gather term pair candidates using both pattern-based and distributional approaches, and then train an SVM classifier to keep the correct pairs. However, many distributional similarity computations are computationally inefficient, because they involve time-consuming sentence parsing to construct dependency trees.

### 3. METHODOLOGY

Firstly, we cast the problem as a conditional probability problem in Section 3.1. Then we talk about how we adapt it for our case in the following sections: we introduce how to get seed synonyms and antonyms in Section 3.2, so that we can automatically construct patterns in Section 3.4. Section 3.5 applies these patterns to perform extraction. We also give a preliminary analysis of synonym and antonym frequencies in the text corpus in Section 3.3.

#### 3.1 Probabilistic Analysis

We demonstrate our probabilistic approach using the example of antonym extraction; synonym extraction is analogous. We define ANT verb pair extraction as finding a set of antonym pairs  $C_{ant}(\mathcal{S})$  in a given text corpus  $\mathcal{S}$ . We only include a candidate verb pair  $VP_i$  in  $C_{ant}(\mathcal{S})$  if the probability  $p(Rel_{ant}|VP_i)$  that it stands in an antonymy relationship, is greater than a threshold  $t$ . Thus we get

$$C_{ant}(\mathcal{S}) = \{VP_i \mid p(Rel_{ant}|VP_i) > t\} \quad (1)$$

Observing the fact that some patterns can indicate antonymy, we introduce pattern based conditional probability to calculate  $p(Rel_{ant}|VP_i)$  indirectly. E.g. pattern “either  $VB_1$  or  $VB_2$ ” is a good indicator, such as “either *live* or *die*”. (We will describe what constitutes a valid pattern in Section 3.4). Assume that we have  $q$  antonym patterns.

$$p(Rel_{ant}|VP_i) = \sum_{j=1}^q (p(Rel_{ant}|Pat_j) \times p(Pat_j|VP_i)) \quad (2)$$

In Equation 2,  $p(Rel_{ant}|Pat_j)$  describes the probability that a pattern  $Pat_j$  can lead to relationship  $Rel_{ant}$ , and the conditional  $p(Pat_j|VP_i)$  indicates the probability that a verb pair  $VP_j$  can co-occur with pattern  $Pat_j$  (See Equations 3 and 4). The product  $p(Rel_{ant}|Pat_j) \times p(Pat_j|VP_i)$  computes the probability that verb pair  $VP_i$  is an antonym pair indicated by a single pattern  $Pat_j$ . The overall probability that a verb pair  $VP_i$  is in relation  $Rel_{ant}$  is the sum of all the probabilities over the  $q$  patterns.

$$p(Rel_{ant}|Pat_j) = \frac{\mathcal{N}_{ant}(j)}{\mathcal{N}(j)}, \quad (3)$$

where  $\mathcal{N}(j)$  is the overall frequency count of  $Pat_j$  and  $\mathcal{N}_{ant}(j)$  is the frequency count of co-occurrence of  $Pat_j$  and  $Rel_{ant}$ . Also,

$$p(Pat_j|VP_i) = \frac{\mathcal{F}_{Pat_j}(i)}{\mathcal{F}(i)}, \quad (4)$$

where  $\mathcal{F}(i)$  is the overall frequency count of  $VP_i$ , and  $\mathcal{F}_{Pat_j}(i)$  is the frequency count of co-occurrence of  $Pat_j$  and  $VP_i$ .

We assume that many patterns are common to our general language use and thus generic enough to be used across different domains. Hence training corpus and testing corpus will share similar distributions of patterns, even if the verbs that participate in the patterns have little overlap. Thus, by equations (3) and (4), we have

$$\begin{aligned} & p(Rel_{ant}|VP_i) \\ &= \sum_{j=1}^n (p^{Test}(Rel_{ant}|Pat_j) \times p^{Test}(Pat_j|VP_i)) \\ &= \sum_{j=1}^n (p^{Train}(Rel_{ant}|Pat_j) \times p^{Test}(Pat_j|VP_i)) \\ &= \frac{1}{\mathcal{F}^{Test}(i)} \sum_{j=1}^n \left( \frac{\mathcal{N}_{ant}^{Train}(j)}{\mathcal{N}^{Train}(j)} \times \mathcal{F}_{Pat_j}^{Test}(i) \right) \end{aligned}$$

All symbols with superscript *Train* refer to the probabilities obtained from the training corpus, and symbols with superscript *Test* are for the testing corpus. In Section 3.4, we introduce how we adapt  $p^{Train}(Rel_{ant}|Pat_j)$  in practice, and in Section 3.5, we show how to compute  $p^{Test}(Pat_j|VP_i)$ .

#### 3.2 Seed Synonym/Antonym Extraction

WordNet[3], a machine readable lexical dictionary, can be taken as a collection of  $\langle f, s \rangle$  pairs, where  $f$  is the *surface form* (for example “go”), and  $s$  stands for *synset*, a set of synonym words, representing a sense out of all the possible senses that a term can convey. For the term “go”, the synonyms “travel, move” belong to the synset that refers to the sense “changing location”, whereas “become, get” belong to another synset expressing the sense of “on the way to another state”. Each synset can also have zero or more corresponding antonyms. The form “stay” can convey either of the following senses: “keep in the old state” and “be in some place”, so its corresponding antonyms are “change” and “move”.

Auxiliary verbs are excluded from seed antonym extraction, because their main purpose is to provide more information about the verbs following them, and they don’t have independent meanings. For example, in the sentence “I *might* come tomorrow.”, “might” does not indicate an action by itself, but modifies the possibility of the coming action. The excluded auxiliary verbs contain: *be, can, do, get, have, may, might, will*, etc. Moreover, to simplify verb recognition, we only extract single verbs instead of compound verb. For all the verbs other than auxiliary verbs, we exhaustively query WordNet and get corresponding antonym verbs.

Based on the above mentioned restrictions, more restriction rules are proposed for seed synonym extraction:

1. Verbs having more than 20 synsets will be removed from the training set. The verb “run” has 41 synsets in WordNet, and 37 synonyms including: *race, work*,

*go, melt, bleed, etc.* Verbs such as *run* will result in over-generalized patterns, since it can be synonym of too many verbs at the same time.

2. If the frequency count of a synset in WordNet is less than 6, the words belonging to this synset will not be extracted. If the frequency count is low, it shows that the words belonging to the synset are rarely used in the sense of the synset.
3. We only extract the top 4 synsets from WordNet. In WordNet, more frequent and popular synsets will be ranked higher. Most of the time, people rarely use the lower rank meanings. For example, the 6th synset of “*light*”, meaning “*alight from (a horse)*”, is rarely used in daily life.

### 3.3 Corpus Analysis

In the LDC English Gigaword Corpus(LDC2005T12)[4], we classify all the verb pairs (excluding auxiliary verbs) with a maximum of 3 words between them into 3 categories: synonym, antonym, and other, based on the seed verb pairs extracted from WordNet. We observe that the average frequency of synonyms is about 40% of that of antonyms, which verifies our claim that people tend to use antonyms more often than synonyms. So it increases the difficulties to extract synonyms.

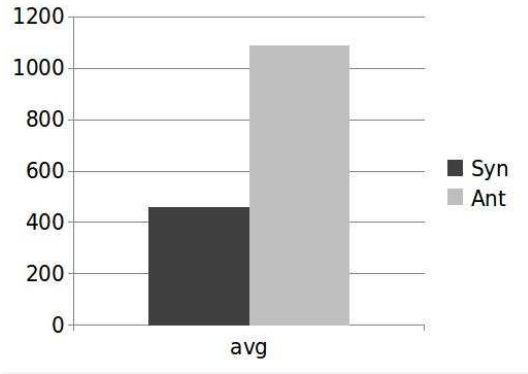


Figure 1: Average frequency of synonym and antonym verb pairs

### 3.4 Pattern Construction

In this section, we talk about defining patterns, constructing a pattern from a sentence, and calculating confidence scores for each pattern. In English, verbs can take the same surface form as nouns or adjectives. Thus, an uninformed extraction algorithm could not distinguish verbs from other parts of speech. For example, “do you *like* the movie?” and “students, *like* Tom work very hard”. The first *like* is used as a verb, while the second one is used as a preposition. The Stanford POS Tagger [11] is used to tag the whole corpus so that we know which are the verbs. Words with POS tags, such as VB, VBD, VBG, VBN, VBP, VBZ, are classified as verbs.

We define a pattern as a consecutive sequence of elements(words and POS tags) of limited length. The SYN or ANT verb pair will be replaced by their corresponding POS tags; other words will keep their surface forms in the original sentence. Then windows with size of 3, 4, and 5 will be

applied to exhaustively construct all the possible patterns. For example, in a parsed sentence “Abacha/NNP can/MD either/RB *accept*/VB or/CC *reject*/VB”, the following patterns will be generated: “<VB> or <VB>”, “either <VB> or <VB>”, “can either <VB> or <VB>”. We chose to keep the tense information for the verb form, because it could be important for the kind of relationship that is represented.

We define  $Rel = \{Rel_{syn}, Rel_{ant}, Rel_{other}\}$ , in which  $Rel$  is the set of relationships,  $Rel_{syn}$  is synonym relationship,  $Rel_{ant}$  is antonym relationship, and  $Rel_{other}$  is relationships other than SYN/ANT. After scanning the whole corpus,  $P_{syn}$ , the set of patterns co-occurring with antonyms can be obtained; similarly, we can get  $P_{ant}$ , and we define  $P$ , the union set of both  $P_{syn}$  and  $P_{ant}$ . We find that many of the patterns can indicate all of the above three relations. Take pattern “<VB> and <VB>” for example, “*maintain and preserve the culture roots*” is a synonym example, “*download and upload files*” is antonym example and “*come and play*” shows non SYN/ANT relationships. Since a pattern can not always indicate a relationship, we use confidence score for every  $\langle P_i, Rel_j \rangle$  pair to indicate the probability that a pattern  $P_i \in P$  leads to a relationship  $Rel_j \in Rel$ .

$$VP2Pattern^{Train} = \begin{bmatrix} f_{1,1} & \cdots & f_{1,n} \\ \vdots & \vdots & \vdots \\ f_{i,1} & f_{i,j} & f_{i,n} \\ \vdots & \vdots & \vdots \\ f_{m,1} & \cdots & f_{m,n} \end{bmatrix} \quad (5)$$

$VP2Pattern^{Train}$  is an  $m$  by  $n$  matrix, representing the co-occurrence count of all verb pairs and patterns, where  $m$  is the total number of verb pairs in training corpus and  $n$  is the size of  $P$ .  $f_{i,j}$  is the number of times that verb pair  $VP_i$  co-occur with pattern  $P_j$ .

$Rel2Pattern$  is a 3 by  $n$  matrix, representing the co-occurrence count of all relationships in  $Rel$  and patterns in  $P$ . For example, we obtain the synonymy row vector by:

$$Rel2Pattern_{R_{syn},j} = \sum_{i \in \{1 \leq i \leq m \mid VP_i \in SYN\}} f_{i,j} \quad (6)$$

Analogous computations create the antonymy vector and the vector that represents other relations.

The first row vector in  $Rel2Pattern$  describes the overall frequency for each pattern co-occurring with all the verb pairs that belong to  $Rel_{syn}$ . And similarly we make the second row vector for relationship  $Rel_{ant}$  and the third row vector for relationship  $Rel_{other}$ .

From the frequency count in the  $Rel2Pattern$  matrix, we acquire  $p(Rel_i|P_j)$ , the conditional probability of a relationship  $Rel_i$  given a pattern  $P_j$ , as shown in Equation 7.

$$Rel2Pattern_{i,j}^{(P|R)} = \frac{Rel2Pattern_{i,j}}{\sum_{j=1}^n Rel2Pattern_{i,j}} \quad (1 \leq i \leq 3) \quad (7)$$

Observing the fact that the frequency counts are more biased towards  $Rel_{other}$ , we boost the rows corresponding to  $Rel_{syn}$  and  $Rel_{ant}$  by row based normalization on  $Rel2Pattern$  in Equation 7. The reason why absolute frequency count in the  $Rel_{other}$  row of  $Rel2Pattern$  are much greater than values in the other two rows is as follows: The number of verb pairs that belong to  $Rel_{other}$  is far greater than that of verb pairs that belong to  $Rel_{syn}$  and  $Rel_{ant}$ . Moreover, we can

only identify those SYN/ANT verb pairs that can be found in WordNet. Other SYN/ANT verbs will initially be misclassified into row  $Rel_{other}$ . Then, column based normalization is applied to  $Rel2Pattern$  so that we get  $p(Rel_i|P_j)$  according to Equation 8.

$$Rel2Pattern_{i,j}^{(R|P)} = \frac{Rel2Pattern_{i,j}^{(P|R)}}{\sum_{i=1}^3 Rel2Pattern_{i,j}^{(P|R)}} \quad (1 \leq j \leq n) \quad (8)$$

Now, after the above mentioned computation (Formula 5, 6, Equation 7, 8),  $Rel2Pattern$  becomes a matrix consisting of cell values ranging from 0 to 1. Each cell  $Rel2Pattern_{i,j}$  shows the probability that pattern  $P_j$  indicates  $Rel_i$ , where  $Rel_i \in \{syn, ant, other\}$ .

### 3.5 Synonym/Antonym Extraction

In Section 3.4, we obtain the confidence values for each pattern indicating a specific relationship from the training corpus. Here, after collecting pattern frequency information from a testing corpus, we will integrate both and extract new synonym and antonym verbs from the testing corpus. Using the same approach mentioned in Formula 5, we collect co-occurring frequency information of verb pairs and patterns from testing corpus, and name it  $VP2Pattern^{Test}$ . Then, the same row based normalization mentioned in Equation 7 is applied, but the row dimension of  $VP2Pattern^{Train}$  is different from that of  $VP2Pattern^{Test}$ . The former is the total number of verb pairs co-occurring with patterns in set  $P$  in training corpus, while the latter is the one in testing corpus. Now,  $VP2Pattern_{i,j}^{Test}$  indicates that given a verb pair  $VP_i$ , what is the probability it will associate with pattern  $P_j$ . We define  $VP2Rel$  by

$$VP2Rel = VP2Pattern^{Test} \times (Rel2Pattern^{Train})^T \quad (9)$$

$(Rel2Pattern^{Train})^T$  is the transpose of  $Rel2Pattern^{Train}$ . Every cell  $VP2Rel_{i,j}$  indicates the probability that the verb pair  $VP_i$  belongs to relation  $j$ . The sum of any row  $i$ ,  $\sum_{j=1}^3 VP2Rel_{i,j}$ , is equal to 1, because any verb pair must be in one of the relations  $syn$ ,  $ant$  or  $other$ .

## 4. CONCLUSION AND FUTURE WORK

In this ongoing work we aim at analyzing inter-verb relationships with high precision. Whereas distributional methods will always have a higher recall than pattern based techniques in this area, their classification will be coarser. Especially, it is difficult to distinguish between synonyms and antonyms, because, as single terms, both tend to occur in similar contexts. Thus, we see this work as complementary to other corpus-based approaches. In future work, we are planning on using LSA-based distributional methods to create candidate sets of highly related verb pairs that we will then analyze more closely with this pattern based technique. We are also investigating optimization techniques that give us a better idea of prior probabilities for patterns and relationships. These priors are difficult to estimate from a restricted corpus, even if the available corpora are fairly large.

## Acknowledgements

This research is supported by research grants from the Human Effectiveness Directorate of the Air Force Research Lab,

WPAFB. Thanks to Ramakanth Kavuluru for several helpful suggestions in the preparation of this document.

## 5. REFERENCES

- [1] T. Chklovski and P. Pantel. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 33–40, Barcelona, Spain, 2004. ACL.
- [2] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, pages 100–110, New York, NY, USA, 2004. ACM.
- [3] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.
- [4] D. Graff, J. Kong, K. Chen, and K. Maeda. English gigaword second edition. 2005.
- [5] M. Hagiwara. A supervised learning approach to automatic synonym identification based on distributional features. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop*, pages 1–6, Columbus, Ohio, USA, 2008. ACL.
- [6] Z. Harris. Distributional structure. In J. J. Katz, editor, *The Philosophy of Linguistics*, pages 26–47. Oxford University Press, 1985.
- [7] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 539–545, Nantes France, 1992. ACL.
- [8] A. Korhonen, Y. Krymolowski, and N. Collier. Automatic classification of verbs in biomedical texts. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 44, page 345, 2006.
- [9] S. Mirkin, I. Dagan, and M. Geffet. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *Proceedings of the COLING/ACL on Main conference poster session*, pages 579–586, Sydney, Australia, 2006. ACL.
- [10] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47, Philadelphia, Pennsylvania, USA, 2002. ACL.
- [11] K. Toutanova, D. Klein, C. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, page 180. Association for Computational Linguistics, 2003.