Extracting Semantically Analogical Word Pairs from Language Resources

Jin Matsuoka Zang Yuling Yves Lepage

Graduate School of Information, Production and Systems, Waseda University
808-0135 Kitakyusyu, Japan
{jinmatsuoka@akane., saber_zyl@fuji., yves.lepage@}waseda.jp

Abstract

A semantic proportional analogy (e.g., sun is to planet as nucleus is to electron) is a pair of word pairs which have similar semantic relations. Semantic proportional analogies may be useful in Natural Language Processing (NLP) tasks and may be applied in several ways. It should have a great potential in sentence rewriting tasks, reasoning systems or machine translation. In this paper, we combine three methods to extract analogical word pairs by using patterns, clustering word pairs and measuring semantic similarity using vector space model. We show how to produce a large amount of clusters exhibiting various quality from good to poor, with a reasonable number of clusters of good quality.

Keywords: Semantic analogy, Contiguity, Similarity, Analogical cluster

1. Introduction

The indexable Web now research at least 11.5 billion pages as of the end of January 2005 (Gulli and Signorini, 2005). It is becoming more and more important to acquire semantic knowledge from the web data using NLP techniques for purposes such as user information, government purpose or ontology building.

As for texts, or pieces of texts down to words, one can examine analogies from two points of view: *formal proportional analogy* and *semantic proportional analogy*. An analogy between four objects A, B, C and D of the same type is usually expressed as follows: "A is to B as C is to D". Formal proportional analogy is between strings of symbols and semantic proportional analogy is between pieces of text considered from their semantic relations or distributional similarity. For example, "walk is to walked as talk is to talked" is a formal proportional analogy and "man is to woman as boy is to girl" is a semantic proportional analogy. Unlike formal analogies of form (Lepage, 2004), solving semantically analogical equations needs and can benefit from a large coverage, machine accessible knowledge source (Akbik and Broß, 2009). The general objective of the research reported here is to build knowledge bases of semantic proportional analogies automatically.

Semantic proportional analogies are analogies on the level of meaning. If two pairs of things (A, B), (C, D) have similar or same semantic relations, they form a semantic proportional analogy. Semantic proportional analogies exist everywhere in our daily life. They were extensively used in SAT problems to test the knowledge of students at university entrance exams in the US.

In this paper, we divide our task into three parts according to the two constitutive notions in analogy and the way they combine to produce semantic proportional analogies. The first part is to extract clusters of word pairs from a corpus by using *contiguity*. The second part is to set up a filter by the property of *similarity* by using vector space model for mapping. The third part combines the previous two parts to produce semantic analogies.

This paper is organized in five sections. After this introduction, we present similar research on proportional analogy in Section 2. In Section 3, we show how to extract semantic analogical word pairs from the contiguity point of view using patterns and the similarity point of view using vector space model. In Section 4, we show how to cluster word pairs and map word pairs. We report our experiments and their evaluation results in Section 5.

2. Related Work

Many researchers proposed various methods to extract proportional analogies. As said above, formal proportional analogy is between strings of symbols and semantic proportional analogy is between words with their meanings. The former was formalized in (Lepage, 2000) under the name of *proportional analogy* in (Lepage, 2004). The definition is based on edit distance between symbols in strings. Proportional analogies are viewed as parallelograms. In the case of "cat is to dog as cats is to dogs", cat is relative to cats as dog is relative to dogs (i.e., singular to plural). In addition, cat is opposed to dog.

The latter, semantic proportional analogies, was studied by (Turney, 2001). Verbal analogies (semantic analogies) are based on relational similarity that can be reduced to attributional similarity (e.g., mason : stone :: carpenter : wood). The basic idea of getting semantic analogies is based on (Gentner and Markman, 1997). Various experiments with 374 multiple-choice SAT word analogy questions using a standard unsupervised machine learning approach, with feature vector based on the frequencies of patterns in a large corpus are described in (Turney, 2005; Turney and Littman, 2005). These papers reported an automatic system with higher performance than that of an average human being. (Bollegala et al., 2009) also proposed a method to compute the similarity between implicit semantic relations in two word pairs. The method for solving SAT word analogy questions consists in clustering semantic relations and in measuring word similarity using Information Theoretic Metric Learning (ITML). Although the precision is 51% on a collection of 374 multiple-choice SAT word analogy questions, it reduces the time taken by (Turney, 2005; Turney and Littman, 2005) from 9 days to less than 6 hours.

3. Semantic Analogy

The reading of an analogical proportion $A : B :: C : D$ between four objects is: A is to B as C is to D. The *is to* and the *as* relations have to be defined. We take *is to* for contiguity and *as* for similarity as suggested by (Itkonen, 2005). In Section 3.1, we show how to build word pairs relying on contiguity. In Section 3.2, we show how to map base word pairs onto target word pairs with similarity.

3.1. Contiguity

Contiguity is the fact of sharing some frontier or boundary for two objects (e.g., fish and fin, or bird and wing). Though contiguity is usually taken to mean *metonymy* in cognitive linguistics, our notion of contiguity is a strong relationship like part-whole relation.

We consider that a word pair shares contiguity, when they can be found in the same sentence and there is some word stream (pattern) between them. This word stream can be defined as a sequences of stop words between contiguous words. For instance, given the following sentence;

> Brand is a piece of **wood** *that has been* **burned** or is burning.

We can extract the contiguous word pair (**wood**, **burned**) with the sequence of stop words: *that has been*.

In contrast, lexical syntactic patterns have been used in various NLP tasks to extract hypernyms (Hearst, 1992; Snow et al., 2004), meronyms (Berland and Charniak, 1999; Girju et al., 2006), synonyms (Davidov and Rappoport, 2006) or paraphrases (Bhagat and Ravichandran, 2008).

In this paper, we use suffix arrays to find sequences of length m in a string of length n. This takes $O(m + \log n)$ time. Suffix arrays can be constructed directly in linear time (Kärkkäinen et al., 2006). We use one of the fastest known suffix array construction algorithms, the SA-IS algorithm (Nong et al., 2009).

3.2. Similarity

Similarity is the fact of sharing some features (e.g., fish and bird, or fin and wing). We build a Vector Space Model (VSM) (Salton et al., 1975) which is an algebraic model for representing any object as a vector of identifiers. The space which we build is made of terms and patterns as Section 3.1. To get term-pattern pairs, we extract the list of stop words on the left and the list of stop words on the right of each non stop word. For instance, given the following two sentences;

> Put the cup on the table.
> Put it on the table.

The words *the*, *on*, *it* are stop words. After scanning the two sentences, we can build the term-pattern matrix M shown in Table 1. We weight each cell value in the vector space model by using Pointwise Mutual Information (PMI):

Table 1: Feature vectors for the words in the text example.

	_ the	the _	_ on the	one the _	_ it on the	it on the _
put	1	0	0	0	1	0
cup	0	1	1	0	0	0
table	0	0	0	1	0	1

$$\mathrm{PMI}(x, y) = \log \frac{p(x, y)}{p(x)p(y)}. \quad (1)$$

Note that we ceil them to zero for negative values. We also remove the noise in the vector space model. We use Singular Value Decomposition (SVD) to induce an approximation space.

$$M = U\Sigma V^T \quad (2)$$

where M is the term and pattern matrix ($n \times m$), U is the term matrix ($n \times k$), Σ is a diagonal matrix of singular values ($k \times k$) and V is pattern matrix ($m \times k$). We can redefine the value k using Formula 3.

$$M \sim \hat{M} = U_k \Sigma_k V_k^T \quad (3)$$

After building the new space $\hat{M}$ according to Formula 3, we measure similarity between words by computing their cosine:

$$\mathrm{cosine}(\hat{M}_i, \hat{M}_j) = \frac{\hat{M}_i \cdot \hat{M}_j}{||\hat{M}_i|| \times ||\hat{M}_j||}. \quad (4)$$

4. Analogical Clustering

Contiguity and similarity are considered in conjunction to obtain clusters of semantically analogical word pairs. We show an example of a semantic analogical cluster in Figure 1. For each word pair in the analogical cluster, there is the same semantic relation of contiguity and they can map by using similarity. From the analogical cluster, we can induce some semantic analogies (three semantic analogies for three word pairs in cluster, i.e., $C_3^2 = 3$).

	Contiguity	
Similarity	length	meter
	volume	liter
	weight	gram

length : *meter* :: *volume* : *liter*
volume : *liter* :: *weight* : *gram*
length : *meter* :: *weight* : *gram*

Figure 1: An example of a semantic analogical cluster (left side) and the semantic proportional analogies represented by this cluster (right side).

To build semantic analogical clusters (*analogical grid*), there are two parts: clustering word pairs with contiguity and mapping by using similarity. In the first part, to build analogical clusters that consist of word pairs with the same semantic relation of contiguity, we use a simple clustering algorithm given in Algorithm 1. This algorithm is simi-

Algorithm 1 Clustering of word pairs

```
  Input: $WPs$ // word pairs
  Output: $clusters$
  while $WPs$ is not empty do
    chose word pairs $wp \in WPs$
    $cluster \leftarrow \{\}$
    $cluster \leftarrow \{wp\} \cup cluster$
    $WPs \leftarrow WPs \setminus \{wp\}$
    for $wp' \in WPs$ do
      if $\text{sim}(wp, wp') \geq \theta$ then
        $cluster \leftarrow \{wp'\} \cup cluster$
        $WPs \leftarrow WPs \setminus \{wp'\}$
      end if
    end for
    if $|Cluster| \geq 2$ then
      $clusters \leftarrow cluster \cup clusters$
    end if
  end while
```

lar to the partitional clustering method. That is, a targeted word pairs is connected to some similar word pairs greedily. In order to connect to some word pairs, we use a Dice coefficient noted sim in the algorithm, and defined as follows:

$$\text{sim}(wp, wp') = \frac{2 \times |\text{ContextSet}(wp) \cap \text{ContextSet}(wp')|}{|\text{ContextSet}(wp)| + |\text{ContextSet}(wp')|}$$

where ContextSet(wp) is the set of unigrams and bigrams of contexts including word pairs (wp) without each word in word pairs. In this way, we get clusters of word pairs which relate in contiguity, not only in similarity.

In the second part, we map a source word pair onto a target word pair in the cluster by using distributional similarity. The Average Similarity Score (AvSS) of a word w in an analogical cluster is defined as follows:

$$\text{AvSS}(w) = \frac{\sum_{w' \in W} \text{sim}(w, w')}{|W|}. \quad (5)$$

Here, w' is on the same side of w in the analogical cluster and w is different from w'. This formula measures the strength for each word on each side in the cluster. That is, the higher the score, the more the word belongs to the given side of the cluster. Words of lower scores are removed. We give an example of calculation in Table 2. Note that the word pairs of UNKOWN label do not exist in the matrix so that we can not process them. After getting AvSS for each word, the next step is to prune the analogical cluster. We define a threshold $Pruning\_Threshold$. For a word w, if AvSS(w) $\leq Pruning\_Threshold$, the word pair which contains w is filtered out. In Table 2, word pairs is deleted if $Pruning\_Threshold \leq 0.2$ (gray cells).

Table 2: The example of cluster and calculation results of formula of AvSS.

$w : w'$	AvSS(w)	AvSS(w')
religion : friend	0.315	0.090
company : famous	0.315	0.090
palatine : famous	UNKNOWN	0.400
knight_errantry : code	UNKNOWN	0.136
titania : queen	UNKNOWN	0.427
oberson : king	UNKNOWN	0.495

5. Experiments and results

5.1. Data and settings

For our experiments, we use two corpora. The first one is the WordNet definition corpus[1]and the second one is the English part of the Europarl corpus[2]. In Table 3, we show some statistics: number of sentences and sentence lengths for each corpus. We show high frequency words handled as stop words to the Table 4 for each corpus. The two corpora are very characteristic because the WordNet definition corpus conforms to patterns of writing but Europarl English corpus is free speech. The WordNet definition corpus is pure written language. The Europarl English corpus is transcribed spoken language. We assume that it is not important to use a large corpus for extracting semantic analogies. If a large amount of patterns appear in a small corpus, good semantic proportional analogies can be extracted. We set the parameters for each part (contiguity, similarity and analogical clustering) for our method in Table 5. In the contiguity part, we define the number of most frequent words for each corpus as 500. To extract word pairs, the substring between words has a size between 4 and 8. In the similarity part, we build the model of terms and patterns for terms occurring more than 20 times in the corpus and patterns consisting of most frequent words where the most frequency words are the top 22 ones. We set the patterns in the matrix with a number of occurrences of more than 50 times in the corpus. To reduce the noise

Table 3: Statistics on two corpora: WordNet definition corpus and Europarl English corpus.

Corpus	# of sentences	sentences length (avg. ± std.)
WordNet defs	133,156	8.94 ± 5.40
Europarl	386,068	28.63 ± 15.48

Table 4: Higher frequent words in each corpus. We use some word in higher frequent words as stop words.

Rank	WordNet defs	freq	Europarl	freq
1	a	68,820	the	770,717
2	of	66,878	,	525,263
3	the	59,239	.	388,031
4	or	39,596	of	363,116
5	in	29,962	to	340,219
6	and	28,066	and	283,696
7	to	25,391	in	237,698
8	)	14,026	that	186,925
9	that	13,352	a	169,676
10	(	13,115	is	164,911

[1] This consists in the definitions of the entries contained in WordNet: `nlpwww.nict.go.jp/wn-ja/index.en.html`

[2] `www.statmt.org/europarl`

Table 6: Experimental results in each corpus.

	# of word pairs	# of clusters	*avg.* cluster size	*avg.* quality score
WordNet definition corpus	13,053	1,905	6.9	3.2
Europarl English corpus	5,803	2,031	2.9	1.7

Table 5: Parameter settings.

Part	Parameter	values
Contiguity	Most frequency words	500
	Substring between words	$4 \sim 8$
Similarity	Terms	$\geq$20
	Most frequency words	22
	Patterns	≥ 50
	Latent variables	60
Clustering	Margining threshold	0.8
	Pruning threshold	0.4

in the matrix by using SVD we define the value of k as 60. Finally, for contiguity clustering, the margining threshold is set to 0.8.

5.2. Experimental Results

Our experimental results are shown in Table 6. We get more word pairs for the WordNet definition corpus than the Europarl English corpus. The number of clusters is almost the same, however, the average cluster size for the WordNet definition corpus is larger than for the Europarl English corpus. This comes from the fact that the WordNet definition corpus contains a lot of repeated patterns but the Europarl English corpus not so much. For evaluation, we assess the quality of clusters by human judgement using a scale from 1 to 5 defined as follows:

1. Totally noise. No contiguity nor similarity can be seen in the cluster;

2. No analogy, but some weak contiguity or similarity can be seen in the cluster;

3. Weak analogy;

4. Good analogy, but a few word pairs in the clusters are noise;

5. Perfect analogy, as good as *teacher* : *to teach* :: *student* : *to study.*

In Table 6, the average quality score of 50 clusters randomly sampled obtained from the WordNet definition corpus is 3.2. The average quality score for 50 clusters randomly sampled in Europarl English corpus is 1.7 and is poor. This indicates that most of the clusters contain noise. These results may be interpreted by the fact that the WordNet definition corpus is much rigid than the Europarl corpus so that the contiguity dimension in the clusters is much better, hence a better overall quality.

We give examples of clusters randomly sampled in each corpus in Table 7 and Table 8.

Table 7: Clusters extracted from the English part of the Europarl corpus with a quality of 4 as judged by human evaluators. The left column consisted of nouns, the right column of past participles or past verbs. The nouns can be seen as the objects of the verbs. In addition, all verbs express the aspect of achievement.

conclusions	:	published
posts	:	approved
evaluations	:	concluded
discussions	:	completed
results	:	announced

6. Conclusion

We proposed a method to extract clusters of semantically analogical word pairs from language resources. The method is divided into three parts: contiguity, similarity and analogical clustering (combining the previous two methods). It relies on the notion of patterns and a pattern mining technique. We performed experiments on two different corpora and gave an analysis of the influence of different corpora of the WordNet definition corpus and the Europarl English corpus. We were able to produce a large amount of clusters of various quality.

7. Acknowledgments

This research was partly supported by “Ambient SoC Global Program of Waseda University” of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

Akbik, Alan and Jügen Broß, 2009. Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *Proceedings of World Wide Web Workshop (WWW)*. Madrid, Spain.

Berland, Matthew and Eugene Charniak, 1999. Finding parts in very large corpora. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL ’99)*:57–64.

Bhagat, Rahul and Deepak Ravichandran, 2008. Large scale acquisition of paraphrases for learning surface patterns. In *Proceeding of Association for Computational Lingusitics (ACL ’08)*, volume 8.

Table 8: Clusters extracted from the WordNet definition corpus with a quality of 5 as judged by human evaluators.

arc_cotangent : cotangent	picofarad : trillionth	lobectomy : lobe	unlawfulness : authorized
arc_tangent : tangent	millifarad : thousandth	mastectomy : breast	unclearness : explicit
arc_cosine : cosine	microfarad : millionth	nephrectomy : kidney	
arc_secant : secant		pneumonectomy : lung	
arc_sine : sine			
arc_cosecant : cosecant			

Bollegala, Danushka T, Yutaka Matsuo, and Mitsuru Ishizuka, 2009. Measuring the similarity between implicit semantic relations from the web. In *Proceedings of the 18th international conference on World wide web*. ACM.

Davidov, Dmitry and Ari Rappoport, 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING '06) and the 44th annual meeting of the Association for Computational Linguistics (ACL '06)*. Association for Computational Linguistics.

Gentner, Dedre and Arthur B Markman, 1997. Structure mapping in analogy and similarity. *American psychologist*, 52:45–56.

Girju, Roxana, Adriana Badulescu, and Dan Moldovan, 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.

Gulli, Antonio and Alessio Signorini, 2005. The indexable web is more than 11.5 billion pages. In *Proceeding of Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM.

Hearst, Marti A, 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics (COLING '92)*, volume 2. Association for Computational Linguistics.

Itkonen, Esa, 2005. *Analogy as structure and process: Approaches in linguistics, cognitive psychology and philosophy of science*, volume 14. John Benjamins Publishing.

Kärkkäinen, Juha, Peter Sanders, and Stefan Burkhardt, 2006. Linear work suffix array construction. *Journal of the Association for Computing Machinery (JACM)*, 53(6):918–936.

Lepage, Yves, 2000. Language of analogy strings. In *Proceedings of the 18th conference on Computational linguistics (COLING 2000)*, volume 1. Stroudsburg, PA, USA.

Lepage, Yves, 2004. Analogy and formal languages. *Electronic notes in theoretical computer science*, 53:180–191.

Nong, Ge, Sen Zhang, and Wai Hong Chan, 2009. Linear suffix array construction by almost pure induced-sorting. In *Data Compression Conference (DCC '09)*. IEEE.

Salton, Gerard, Anita Wong, and Chung-Shu Yang, 1975. A vector space model for automatic indexing. *Communications of the Association for Computing Machinery (ACM '75)*, 18(11):613–620.

Snow, Rion, Daniel Jurafsky, and Andrew Y. Ng, 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*.

Turney, Peter, 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. *In Proceedings of the Twelfth European Conference on Machine Learning*:491–502.

Turney, Peter D., 2005. Measuring semantic similarity by latent relational analysis. *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI '05)*:1136–1141.

Turney, Peter D and Michael L Littman, 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278.