Discovering Analogous Pairs of Words using WordNet and PageRank

First Author
Affiliation / Address line 1
Affiliation / Address line 2
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
email@domain

Abstract

We present an unsupervised algorithm for identifying analogous pairs of concepts based on corpus statistics and PageRank. *Some sentence here on the general context of the paper.* The algorithm uses context to identify word pairs that are most likely to share the same relation. To move from words to concepts, for each grouping of analogous word pairs we build a graph of all the possible senses and apply the PageRank algorithm to determine which are most likely. We test how analogous are the resulting lists of concepts with the SAT analogy questions used in (Turney et al., 2003). *Some closing sentence here about what we found out based on the results.*

1 Introduction

Humans are adept at discovering analogous relationships between pairs of concepts. The concepts may be closely related such as "fish swim in the sea; birds fly in the sky," or unrelated "electrons orbit protons; plants orbit stars." There has been much work on how computers might emulate this **discovery process**(Gentner, 1983). *Either cite a bunch of things here, or add a short setence or two describing the work.* Furthermore, recent work has show how text-only based approaches can identify analogous words or concepts(Mangalath et al., 2004; Turney and Littman, 2005; Biçici and Yuret, 2006). We propose a novel algorithm for identifying analogous pairs of concepts by combining at context statistics for word usage and PageRank(Brin and Page, 1998) to determine what concepts those words represents.

This paragraph should outline the specifics of what problem is addressed in this paper. This paragraph must catch the reader/reviewrs attention and make them believe the problem is important.

A significant challenge with generating analogies directly from text is that standard english writing does not frequently contain phrases such as "*a* is to *b* as *c* is to *d*". Another which occurs in writings is a lack of comparable contexts, such that if you are given a $noun\_verb\_noun$ triple, there are few occurances of $*\_verb\_noun$ and $noun\_verb\_*$, and even fewer occurances of $noun\_otherverb\_noun$. An important contribution of this paper is the use of the Open Mind Common Sense (Havasi et al., 2007) collection of common sense facts.

Additionally, we handle the issue of many potential meanings for words by combining new approaches to Word Sense Disambiguation, which not only provides more accuate relations, but can also reduce the scope of our problem.

1. Writing does not often contain explicit phrases of the type "*a* is to *b* as *c* is to *d*"
2. Even when we look for words that appear in simiar context, there may not be many occurrences.
3. words can take on a number of meanings, so a solution must be more than just extraction context-related words

Next, we outline our proposed solution and how we address the problems in the previous paragraph. We should give a motivating example here

Our hypothesis is that Pairs of words that appear in similar semantic contexts are likely to share some relation to each other. In this paper we focus specifically on pairs of nouns that are likely to share a relation when the pairs or their synonyms occur in the presence of verbs with similar semantics.

This could be a footnote: Note that our approach does not extract what the specific relationship is between concepts. Determining the relation is a task of language generation and could require significant background knowledge for complex relationships, both of which is outside the current scope of the task.

Then we need to frame our work in the existing body of work and state how we will measure up to prior examples. Stating what we can use the work for is also a plus.

The general outline of our paper is as follows: Bob Lob Law.

2 Background

Our analogy discovery system makes use of two well established databases of knowledge, WordNet (Fellbaum, 1998) and the Open Mind Common Sense (Havasi et al., 2007) collection. Both of these provide suitable search spaces for analogous noun-noun pairs, and a means of putting a reasonable scope to the problem.

Wordnet (Fellbaum, 1998) is a commonly used Lexical database, primarily used for Word Sense Disambiguation. For all open class words, such as nouns and verbs, WordNet contains a collection of synsets which represent possible meanings of a particular word. For instance, the word car could refer to an automobile, typically with 4 wheels, or it could also refer to a wheeled vehicle for use on railroad lines. Each of these possible meanings is reprensented as a synset, which encompasses a collction of words which have the same meaning.

WordNet is further organized to include various relations between synsets, the most important of which are: Hypernym, which provides a more general synset; and Hyponym, which provides a more specific synset. Additionally WordNet contains relations for similar-to relations, meronymy, holonymy, and more, but we primarily focus only on hyponyms and hypernyms. Traditionally this is called the IS-A hierarchy.

While WordNet is an excellent resrouce for WSD, it alone does not provide enough information to generate analogies. For the purpose of extracting potential noun-noun pairs which can participate in an analogy, we make use of the Open Mind Common Sense. The facts in OMCS are written in a simplified version of english which provides a vast amount of facts and relations between concepts. These facts are primairly collections of attributes for a particular topic, such facts as: “paper is white“‘, “paper is for writing on“, etc. Due to the simplified english, and wide range of provided relations, OMCS appears to be an excellent resource for analogy generation, and has been used in similar projects (Speer et al., 2008). From OMCS, we make use of approximately 130,000 $noun\_verb\_noun$ triplets, which is a reasonable sized corpus for discovering groups of related noun-noun pairs.

Initially, other corpora were explored in the hopes of extracting suitable analogies. A variety of books from Project Gutenberg were explored, Moby Dick, Childrens Novels, Paradise Lost, etc. Unfortunately, due to the advanced use of english in the majority of these texts we ran into several problems. The first problem encountered as a lack of sensible $noun\_verb\_noun$ triplets from the corpora. Second, we were unable to find additional examples of triplets based on synonyms of nouns and verbs, which left a remarkable small search space for real world examples, leaving the discovery of analogies to traversal of WordNet relations, which is perhaps beyond the scope of WordNet’s intended usage.

3 Identifying Analogous Pairs

Insert general introductory paragraph that oulines the algorithm and how it will be broken up and explained.

We are interested in determining if there exists a relationship between a pair of nouns, which is dependent on identifying the semantic context in which the nouns are used. We use the verb as the key indicator of context and a good starting point for identifying a possible noun-noun relation. The general theme of the algorithm is that two pairings of nouns will share a relation when the nouns or their synonyms occur in the presence of verbs with similar semantics. Consider the following sentences:

1. The thisty man guzzled the water.

2. The old car guzzled gas quickly.

In a simple case such as this, the relation between man and water is the same as the relation between car and gas. It might also be true that the car, gas relation mighthold to other things which a man might guzzle, or beings which might guzzle water. To account for this, our approach is

to expand a potential noun-noun pair by searching the corpus itself for other noun-noun pairs which might have a similar meaning according to context.

This process takes place in three major steps

1. Identify candidate noun-noun pairs
2. Expanding each noun-noun pair
3. Remove expanded pairs which are unrelated
4. Categorize the expanded pairs for comparison

The following subsections will cover each step is more detail.

3.1 Generating Candidate Analogies

This paragraph should focus on the first step $noun\_verb\_noun$ extraction from the corpus. We allow that nouns may be separated from the verb by one or more word of non-noun parts of speech. We also filter out the nouns in prepositional phrases. Last, we allow nouns to be collocations, e.g. "white house".

3.2 Moving From Word to Concept

Once a list of $n_1\_v\_n_2$ triplets have been extracted from the corpus, we need to expand each one to it's set of extended relations. The first step is to search the corpus for other triplets which match the pattern of $n_1\_v\_*$ and $*\_v\_n_2$. This will find triplets which contains words which might be viewed as synonymous in some way and is loosely related to finding Selectors for WSD (Schwartz and Gomez, 2008). An additional way to generate additional relations is to produce a new triplets according to:

$n_1\_v\_n_{2i} \forall n_{2i} \in synonym(n_2)$

and

$n_{1i}\_v\_n_2 \forall n_{1i} \in synonym(n_1)$.

Since we are making use of WordNet as our definition of word meaning, it is most applicable to extract synonyms from the dominate sense of each word in the triplet.

This generation step has the potential of creading a vast number of $noun\_verb\_noun$ triplets which we consider to be synonomous to each other, and a simple, unsupervised approach needs to be taken to reduce the size of the expanded list for later steps. To this end, we chose to make use of PageRank when applied to WSD. The rest of this section will focus on how the graph is built, and how it's computation reflects on the expansion list.

Our use of PageRank is an extension of it's application towards WSD (**?**) and (Mihalcea, 2006). For each word in the expansion list, we obtain the first three possible synsets for the word and add edges from each competeing synset to all other synsets currently in the graph based on the semantic similarity as defined by (Banerjee and Pedersen, 2003). One detail to note, is that competing synsets for a given word are only added to the graph onces, regardless of the number of times it occurs in the extension list, and these competing synsets are not given edges to each other.

The semantic similarity metric defined by (Banerjee and Pedersen, 2003) gives a score based on the gloss overlap of two synsets, along with gloss overlap of their respective Hyponyms and hypernyms. Their approach can be extended to take into account additional WordNet relations, but we only evaluate the hyponym and hypernym glosses in addtion to the synset glosses themselves. The formulate used is as follows:

$$similarity(A,B) = score(gloss(A), gloss(b)) + score(hyper(A), hyper(B)) + score(hypo(A), hypo(B)) + score(hyper(A), gloss(B)) + score(gloss(A), hyper(B))$$

Where the score function is a sum of the number of words overlapping in the two definitions, giving a higher weight to longer sequences of words which match. For the hypernym and hyponym relations, we take only the first hypernym or hyponym synset given by wordnet, as opposed to concatening each of their glosses together.

Once the graph has been built with all the possible sense of each word that occurs in the expansion list, an arbitrary score is assigned to each node in the graph. This score will represent the importance of the particular synset, such that a higher score implies that the synset is more fitting for the word given the context, and conversely a lower score implies that the synset would not be a fitting sense. Note that since PageRank will rely on converging the score for each node, the initial scores do not affect the final value, instead they only affect the number of iterations needed for convergence. For simplicity, we select initial values of 1 for every node.

Since our graph assigns edge weights to every edge between nodes, and each edge is un directed, we iteratively update the scores for each synset with algorithm as follows, which is adapted from

(Mihalcea, 2006).

$$PR(V_i) = (1-d) + d * \sum_{V_j \in Edge(V_i)} w_{ij} \frac{PR(V_j)}{\sum_{V_k \in Edge(V_j)} w_{jk}}$$

This update will run until for each node V_i the difference $PR^{K+1}(V_i) - PR^K(V_i)$, changes less than some threshold, where ours is 10^{-6}. Typically, this can be done in lesss than 40 iterations through the graph.

The final step utilizing PageRank and the expansion list s filtering out $noun\_verb\_noun$ triplets which are not semanticly related to the others. Each word that occured in the expansion list is then given a score according to the number of synsets for the word which has a pagerank score above some threshold t. Then, each $noun\_verb\_noun$ triplet is given a score based on the following equation:

$$PairScore(nvn) = \frac{\sum_{w \in NounsOf(nvn)} score(w) * count(w)}{\sum_{x \in UniqueWord(expansionList)} count(x)}$$

3.3 Moving from Concepts to Categories

We may not get to trying this One thought I had was to try to cluster concepts from within the list itself. We would still maintain the invariant that all pairs are still analogous, but this would at least give us:

1. a way of moving from concepts to categories, i.e. are there certain pairs of categories that are analogous?

2. if we can extract categories, then we might be able to better identify new analogies. For example, if we find that we have the categorical analogy "animal:type" and we see narwhal, we know it fits in the animal category even if we have never see "narwhal" or any of its synonyms.

4 Evaluation

The important evaluation criteria is whether all the pairs in a generated set of analogous pairs actually share the same relation. We test this criteria using SAT analogy questions of the form shown in figure 1. A question will provide an exemplar pair with a certain relation and a list of possible analogous pairs; the correct answer requires identifying the option with the pair that shares the same relationship. For evaluation we use the 374 SAT analogy problems used in (Turney et al., 2003), which provide five options to select from.

Provided example:		mason:stone
Option	(a)	teacher:chalk
	(b)	carpenter:wood
	(c)	soldier:gun
	(d)	photograph:camera
	(e)	book:word
Solution	(b)	carpenter:wood

Figure 1: An example SAT analogy question. Example reproduced from (Turney et al., 2003).

4.1 Rationale

Our algorithm for selecting which option is summarized as choosing the set that contains both the provided pair and the selected option, with possibilities for substituting words in each pair with their synonyms. We have purposefully kept the algorithm for answering SAT analogy questions simple to focus primarily on on evaluating the validity of the relationship within the sets and not on our ability to correctly answer all the questions. We find the information retrieval concepts of precision and recall useful in this evaluation; we define precision as a function of how well our algorithm does at producing a set with a specific relationship, and recall as a function of the corpus. If the algorithm groups pairs with too broad of a relationship or no relationship at all, then selecting the correct option from an SAT question will generate multiple false positives. However, if the corpus from which the sets are extracted does not contain enough exemplars of a relationship, then the sets will be impoverished and given an analogy question, no set will contain both the provided example and one of the options. Therefore, we provide two statistics, one for questions for which the algorithm was able to find a set containing both the example and an option (precision), and a second for which we were unable to find such a set.

4.2 SAT Question Algorithm

Our algoirthm for solving SAT-style analogy questions is as follows. *reword this all later. This might be better presented using paragraphs*

1. sets $S_1 \ldots S_n$ of analogous pairs are generated by the algorithm

2. the input is a pair of words denoted $a : b$

   First determine which set contains pairs that have the same relation as the input.

3. for each synonym of a and b, generate a new pair $a' : b'$.

4. find all pairs in the provided sets that match $a_i : y$ and $x : b_i$. Let the set of mathes be M.

5. for each pair $a_i : b_j \in M$ compute the Banerjee distance from a to a_i and b to b_i and find set that contains the pair with the lowest sum. Let the resultings set me known as X *need a better name*. Note that if the input pair is in the provided sets, the distance will be 0 and so the set containing the input pair will be used.

6. If $|X| = 0$, return that no analogy could be found

   Second determine which option has the analogous relation.

7. for each option, $c : d$, generate a new pair $c' : d'$ based on the synonyms of c and d.

8. find all pairs in X that are of the form $c_i : y$ or $y : d_i$, and let this set be N

9. for each pair $c_i : d_i \in N$, find the pair with the lowest Banerjee distance to the original option pair from which it was generated. (**What to do in case of ties? Idea: use the option that is deepest in the tree?**)

10. If $|N| = 0$ return that no analogy coudl be found. Otherwise, select the option with the lowest distance.

These Banerjee distance used in Steps ? and ? are weighting factors for the semantic distance from the provided analogous pair. We use these to account for any differences in semantics when searching for the closest pair in the provided sets

We allow synonyms of the all the pairs to account for some missing features in the corpus. *more details and example here.*

4.3 Results

I sure hope these are good...; Save the analysis until the discussion section

5 Related Work

Mangalath et al. extend LSA(Landauer et al., 1998) with a way determine relation categories for word pairs(Mangalath et al., 2004). The authors define ten types of relations and words that are describe the relations, e.g. the relation synonymy and desciptors "equivalent," "equal," and "match." For each of these ten relations, the authors use LSA vectors to compute the cosine similarity between the words and each provided pair and each alternative. The alternative whose similarities are most correlated with the provided pair is selected as the answer.

Nakov and Hearst indentify the semantic relation between compound noun by seeing what verbs might be used to related them(Nakov and Hearst, 2006). Given some pair of words that constitutes a compound noun, they search web pages for short phrases where the two nouns are linked by a verb. The types of verbs found indicate the type of relationship between the nouns. Unlike this work, no attempt is made to automatically group noun pairs according to the relations they share.

Silva et al. use Bayesian statistics to solve the problem of finding new pairs that share a relation, given some example pairs that are known to share the same relation(Silva et al., 2007). Their approach differs in that which concepts are related is provided as background knowledge.

Speer et al. also use the Open Mind Common Sense knowledge base (KB) to find analogies(Speer et al., 2008). Natural language assertions are redefined as a concept and a features; for example "a trunk is a part of a car," is turned into the concept"trunk" with the feature "part-of(car)." The entire KB is used to construct the matrix of concept $\times$ feature. This matrix is sparse due to missing data, and also may contain erroneous features due to incorrect assertions in the KB. Therefore, in order to smooth out the data, the matrix is refactored using Singular Value Decomposition, keeping only a few dimensions, $\text{SVD}_k(A) = U_k S_k V_k^\top$. Concepts are represented as rows in the U_k matrix, and analogous concepts can be found by looking for those concept vectors whose cosine similarity is highest.

Turney et al. combine several techniques for solving analogy questions and use a weighting mechanism to achieve a better total accuracy than any of the individual techniques alone(Turney et

al., 2003). The authors include common techniques for comparing word similarity such WordNet relations (e.g. hypernyn, meronym) and gloss overlap. Another notable technique, the authors express the realtionship between two words X and Y as a 128 element vector; the elements are the frequencies returned by a search engine query of the form "X P Y", where P is one of 128 predetermined relation-like phrases such as "for," "with," or "in the." These phrases provide a closed-set of ways of relating the nouns; in comparison, our approach uses verbs to relate, which makes a vector-like representation impossible due to the open-class nature of verbs. In (Turney and Littman, 2005), Turney and Littman use a similar vector representation for words to solve analogy questions.

Biçici and Yuret extend the work by (Turney and Littman, 2005) by clustering the vector representations of words(Biçici and Yuret, 2006). The authors compute multiple cluster segmentations using k-means and spectral clustering. Analogy questions are then answered by counting how many clusters contain both the query pair and each option; the option that occurs most frequently with the querty pair in the clusters is selected as the answer. **mention that they do worse than Turney?**

Veale also uses gloss overlap and the hypernym trees from WordNet as a part of solving analogy questions(Veale, 2004). Word sense similarity is computed using a combination of exponentially-weighted gloss overlap and depth of the common hypernym ancester of both senses. Analogy questions are solved by finding the choice pair with the highest similarity to the provided pair.

6 Discussion

Hopefully we have good results and this section can focus on putting them in context. Key questions we can address are

1. Why does this approach work?
2. Are there cases where it doesn't work?
3. Could we improve the SAT question answering algorithm?
4. How does our model fit in with the rest of the field. This point should draw upon all of the related work to paint a picture of the research landscape. Can we combine our approach with others (e.g. use us as input for Sivla et al.), or adapt some techniques of other's papers?
5. Aren't we just taking advantage of the knowledge contained in corpus (to some degree, yes)
6. How we can improve this approach, i.e. future work. One possibility is to move to analogous verbs, adjectives. A second is to adapt the triplet extractor to be more general. A third is to try to move to new specific domains, e.g. mine medical journal papers/abstracts to find analogous things.

7 Acknowledgements

We should probably thank Peter Turney for the SAT Questions and Catherine Havasi for giving us the OMCS data set. Also, if we do, due to the double-blind reviewing process we can't include it in the paper until the print copy since it gives it away that we're not them.

References

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810.

E. Biçici and D. Yuret. 2006. Clustering word pairs to answer analogy questions. In *Proceedings of the Fifteenth Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN 2006)*.

S. Brin and L. Page. 1998. The anatomy of a large-scale hyper-textual web search engine. *Computer Networks and ISDN Systems*, 30(1-7).

C. Fellbaum. 1998. Wordnet, an electronic database.

D. Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2).

C. Havasi, R. Speer, and J. Alonso. 2007. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Proceedings of Recent Advances in Natural Languges Processing*.

T. Landauer, P. W. Foltz, and D. Laham. 1998. Introduction to latent semantic analysis. *Discourse Processes*, (25):259–284.

P. Mangalath, J. Quesada, and W. Kintsch. 2004. Analogy-making as predication using relational information and lsa vectors. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*.

R. Mihalcea. 2006. Random walks on text structures. *Computational Linguistics and Intelligent Text Processing*, 3878:249–262.

Preslav Nakov and Marti Hearst. 2006. Using verbs to characterize noun-noun relations. In *Artificial Intelligence: Methodology, Systems, and Applications*, volume 4183.

H. Schwartz and F. Gomez. 2008. Acquiring knowledge from the web to be used as selectors for noun sense disambiguation. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 105–112, Manchester, England, August. Coling 2008 Organizing Committee.

R. Silva, K. Heller, and Z. Ghahramani. 2007. Analogical reasoning with relational bayesian sets. In *11th International Conference on Artificial Intelligence and Statistics*.

Robert Speer, Catherine Havasi, and Henry Lieberman. 2008. Analogyspace: Reducing the dimensionality of commonsense knowledge. In *Conference of the Association for the Advancement of Artificial Intelligence (AAAI-08)*, Chicago, July.

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

P.D. Turney, M.L. Littman, J. Bigham, and V. Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, pages 482–489.

T. Veale. 2004. Wordnet sits the sat: A knowledge-based approach to lexical analogy. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 606–612.