A Dependency Graph Isomorphism for News Sentence Searching

Kim Schouten and Flavius Frasincar

Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands
{schouten,frasincar}@ese.eur.nl

Abstract. Given that the amount of news being published is only increasing, an effective search tool is invaluable to many Web-based companies. With word-based approaches ignoring much of the information in texts, we propose Destiny, a linguistic approach that leverages the syntactic information in sentences by representing sentences as graphs with disambiguated words as nodes and grammatical relations as edges. Destiny performs approximate sub-graph isomorphism on the query graph and the news sentence graphs, exploiting word synonymy as well as hypernymy. Employing a custom corpus of user-rated queries and sentences, the algorithm is evaluated using the normalized Discounted Cumulative Gain, Spearman's Rho, and Mean Average Precision and it is shown that Destiny performs significantly better than a TF-IDF baseline on the considered measures and corpus.

1 Introduction

With the Web continuously expanding, humans are required to handle increasingly larger streams of news information. While skimming and scanning can save time, it would be even better to harness the computing power of modern machines to perform the laborious tasks of reading all these texts for us. In the past, several approaches have been proposed, the most prominent being TF-IDF [6], which uses a bag-of-words approach. Despite its simplicity, it has been shown to yield good performance for fields like news personalization [1]. However, the bag-of-words approach does not use any of the more advanced linguistic features that are available in a text (e.g., part-of-speech, parse tree, etc.).

In this paper we propose a system that effectively leverages these linguistic features to arrive at a better performance when searching news. The main idea is to use the dependencies between words, which is the output of any dependency parser, to build a graph representation of a sentence. Then, each word is denoted as a node in the graph, and each edge represents a grammatical relation or dependency between two words. Now, instead of comparing a set of words, we can perform sub-graph isomorphism to determine whether the sentence or part of a sentence as entered by the user can be found in any of the sentences in the database. Additionally, we implemented the simplified Lesk algorithm [4] to perform word sense disambiguation for each node, so that it will represent the word together with its sense.

The method we propose to compare to graphs is inspired by the backtracking algorithm of McGregor [5], but is adjusted to cope with partial matches. The latter is necessary since we do not only want to find exact matches, but also sentences that are similar to our query to some extent. As such, we aim to produce a ranking of all sentences in the database given our query sentence.

2 News Searching

To compare two graphs, we traverse both the query sentence graph and each of the news sentence graphs in the database in a synchronized manner. Given a pair of nodes that are suitable to compare, we then recursively compare each dependency and attached node, assigning points based on similarity of edges and nodes. In this algorithm, any pair of nouns and any pair of verbs is deemed a proper starting point for the algorithm. Since this results in possibly more than one similarity score for this news-query sentence combination, we only retain the highest one.

The scoring function is implemented as a recursive function, calling itself with the next nodes in both the query graph and the news item graph that need to be compared. In this way, it traverses both graphs in parallel until one or more stopping criteria have been met. The recursion will stop when there are either no more nodes or edges left to compare in either or both of the graphs, or when the nodes that are available are too dissimilar to justify comparing more nodes in that area of the graph. When the recursion stops, the value returned by the scoring function is the accrued value of all comparisons made between nodes and edges from the query graph and the news item graph.

A genetic algorithm has been employed to optimize the parameters that weigh the similarity score when comparing nodes and edges. Mainly used to weigh features, an additional parameter is used to control the recursion. If there is no edge and node connected to the current node that is able to exceed this parameter, the recursion will stop in this direction.

Computing the similarity score of edges is simply done by comparing the edge labels, which denote the type of grammatical relation (e.g., subject, object, etc.). For nodes, we compare five word characteristics: stem, lemma, literal word, basic POS category (e.g., noun, verb, adjective, etc.), and detailed POS category (plural noun, proper noun, verb in past tense, etc.). These lexico-syntactic features are complemented by a check on synonymy and hypernymy using the acquired word senses and WordNet [2]. Last, by counting all stems in the database, we adjust the node score to be higher when a rare word rather than a regular word is matched.

3 Evaluation

In this section, the performance of the Destiny algorithm is measured and compared with the TF-IDF baseline. To that end, we have created a database of 19 news items, consisting of 1019 sentences in total, and 10 query sentences. All possible combinations of query sentence and news sentence were annotated by at least three different persons and given a score between 0 (no similarity) and 3 (very similar). Queries are constructed by rewriting sentences from the set of news item sentences. In rewriting, the meaning of the original sentence was kept the same as much as possible, but both words and word order were changed (for example by introducing synonyms and swapping the subject-object order). The results are compared using the normalized Discounted Cumulative Gain (nDCG) over the first 30 results, Spearman's Rho, and Mean Average Precision (MAP). Since the latter needs to know whether a result is relevant or not, and pairs of sentences are marked with a score between 0 and 3, we need a cut-off value: above a certain similarity score, a result is deemed relevant. Since this is a rather arbitrary decision, the reported MAP is the average MAP over all possible cut-off values with a step size of 0.1, from 0 to 3.

3.1 Quality of search results

In order to assess our solution's performance, it is compared with a TF-IDF baseline on three measures. Each of the measures is computed using the user-rated sentence pairs as the golden standard. Table 1 shows the results of all three tests, clearly demonstrating that Destiny significantly outperforms the TF-IDF baseline. The p-value for nDCG and Spearman's Rho is computed for the paired one-sided t-test on the two sets of scores consisting of the 32 split scores for both Destiny and TF-IDF, respectively. For MAP, because we computed the average over all cut-off values, the same t-test is computed over 30 cut-off values × 32 folds which results in 960 split scores.

Table 1. Evaluation results

	TF-IDF mean score	Destiny mean score	rel. improvement	t-test p-value
nDCG	0.238	0.253	11.2%	< 0.001
MAP	0.376	0.424	12.8%	< 0.001
Sp. Rho	0.215	0.282	31.6%	< 0.001

4 Concluding Remarks

Our implementation of Destiny shows the feasibility of searching news sentences in a linguistic fashion, as opposed to using a simple bag-of-words approach. By means of a natural language processing pipeline, both news items and queries are processed into graphs, which are subsequently compared to each other, with the degree of sub-graph isomorphism as a proxy for similarity. Because this graph-representation preserves much of the original semantic relatedness between words, the search engine is able to utilize this information. Furthermore, words are not only compared on a lexico-syntactic level, but also on a semantic level by means of the word senses as determined by the word sense disambiguation implementation. This also allows for checks on synonymy and hypernymy between words. Last, the performance results on the Mean Average Precision, Spearman's Rho, and normalized Discounted Cumulative Gain demonstrate the significant gain in search results quality when using Destiny compared to TF-IDF.

Interesting topics for future work include the addition of named entity recognition and co-reference resolution to match multiple referrals to the same entity even though they might be spelled differently. Our graph-based approach would especially be suitable for an approach to co-reference resolution like [3], as it also utilizes dependency structure to find the referred entities.

Acknowledgment

The authors are partially supported by the Dutch national program COMMIT.

References

1. J. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn. Open User Profiles for Adaptive News Systems: Help or Harm? In *16th International Conference on World Wide Web (WWW 2007)*, pages 11–20. ACM, 2007.
2. C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.
3. A. Haghighi and D. Klein. Coreference Resolution in a Modular, Entity-Centered Model. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2010)*, pages 385–393. ACL, 2010.
4. A. Kilgarriff and J. Rosenzweig. English senseval: Report and results. In *2nd International Conference on Language Resources and Evaluation (LREC 2000)*, pages 1239–1244. ELRA, 2000.
5. J. J. McGregor. Backtrack Search Algorithms and the Maximal Common Subgraph Problem. *Software Practice and Experience*, 12(1):23–34, 1982.
6. G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.