Matching sets of parse trees for answering multi-sentence questions

Boris Galitsky
Knowledge Trail Inc. San Jose USA
bgalitsky@hotmail.com

Dmitry Ilvovsky, Sergey Kuznetsov, Fedor Strok
Higher School of Economics, Moscow Russia
dilv_ru@yahoo.com;
skuznetsov@hse.ru;
fdr.strok@gmail.com

Abstract

The problem of answering multi-sentence questions is addressed in a number of products and services-related domains. A candidate set of answers, obtained by a keyword search, is re-ranked by matching the set of parse trees of an answer with that of the question. To do that, a graph representation and learning technique for parse structures for paragraphs of text have been developed. Parse Thicket (PT) as a set of syntactic parse trees augmented by a number of arcs for inter-sentence word-word relations such as co-reference and taxonomic relations is introduced. These arcs are also derived from other sources, including Speech Act and Rhetoric Structure theories. The proposed approach is subject to evaluation in the product search and recommendation domain, where search queries include multiple sentences. An open source plugin for SOLR is developed so that the proposed technology can be easily integrated with industrial search engines.

1 Introduction

Modern search engines are not very good at tackling queries consisting of multiple sentences. They either find very similar documents, if they are available, or very dissimilar ones, so that search results are not very useful to the user. This is due to the fact that for multi-sentences queries it is rather hard to learn ranking based on user clicks, since the number of longer queries is practically unlimited. Hence we need a linguistic technology, which would rank candidate answers based on structural similarity between the question and the answer. In this study we build a graph-based representation for a paragraph of text so that we can track the structural difference between these paragraphs, taking into account not only parse trees, but the whole discourse as well.

Paragraphs of text as queries appear in the search-based recommendation domains (Montaner et al., 2003; Bhasker and Srikumar 2010; Thorsten, 2012). Recommendation agents track user chats, user postings on blogs and forums, user comments on shopping sites, and suggest web documents and their snippets, relevant to a purchase decisions. To do that, these recommendation agents need to take portions of text, produce a search engine query, run it against a search engine API such as Bing or Yahoo, and filter out the search results which are determined to be irrelevant to a purchase decision. The last step is critical for a sensible functionality of a recommendation agent, and poor relevance would lead to a lost trust in the recommendation engine. Hence an accurate assessment of similarity between two portions of text is critical to a successful use of recommendation agents.

Parse trees have become a standard form of representing the syntactic structures of sentences (Abney, 1991; Punyakanok *et al.*, 2005; Domingos and Poon, 2009). In this study we will attempt to represent a linguistic structure of a paragraph of text based on parse trees for each sentence of this paragraph. We will refer to the set of parse trees plus a number of arcs for inter-sentence relations between nodes for words as *Parse Thicket* (PT). A PT is a graph, which includes parse trees for each sentence, as well as additional arcs for inter-sentence relationship between parse tree nodes for words.

We define the operation of generalization of text paragraphs via generalization of respective PTs to assess similarity between them. The use of generalization for similarity assessment is inspired by structural approaches to machine learning (Mill, 1843; Mitchell, 1997; Furukawa 1998; Finn, 1999) versus statistical alternatives where similarity is measured by a distance in feature space (Fukunaga, 1990; Manning and

Schütze, 1999; Byun and Lee, 2002; Jurafsky and Martin, 2008). Our intention is to extend the operation of least general generalization (e.g., the antiunification of logical formulas (Robinson, 1965; Plotkin, 1970)) towards structural representations of paragraph of texts to compute similarity between multi-sentence questions and answers. Hence we define the operation of generalization on a pair of PT as finding the maximal common sub-thickets based on generalizing phrases from two paragraphs of text.

Generalization of text paragraphs is based on the operation of generalization of two sentences, explored in a few studies (Galitsky *et al.*, 2008; Galitsky, 2012). In addition to learning generalizations of individual sentences, we learn how the links between words in sentences other than syntactic ones can be used to compute similarity between texts. We rely on our formalizations of the theories of textual discourse such as *Rhetoric Structure Theory* (Mann et al., 1992) to improve the ranking of paragraph-based question answering.

Whereas machine learning of syntactic parse trees for individual sentences is an established area of research, the contribution of this paper is a structural approach to learning syntactic information at the level of paragraphs. A number of studies applied machine learning techniques to syntactic parse trees (Collins and Duffy, 2002), convolution kernels (Haussler, 1999) being the most popular approach (Lodhi *et al.*, 2002; Moschitti, 2006; Zhang *et al.*, 2008, Zhang *et al.*, 2008, Sun *et al.*, 2010).

2 Parse Thickets for matching questions and answers

Once we have a sequence of parse trees for a question, and that of an answer, how can we match these sequences? A number of studies compute pair-wise similarity between parse trees (Collins and Duffy, 2002; Punyakanok *et al.*, 2003; Moschitti, 2006). However, to rely upon discourse structure of paragraphs, and to avoid dependence of how content is distributed through sentences, we represent the whole paragraphs of questions and answers as a single graph and call it Parse Thicket (PT). To determine how good is an answer for a question, we match their respective PTs.

We extend the syntactic relations between the nodes of the syntactic dependency parse trees towards more general text discourse relations. Once we have such relations as "the same entity", "sub-entity", "super-entity" and anaphora, we can extend the notion of phrase to be matched between texts. In case of single sentences, we match noun, verb, and other types of phrases in questions and answers. In case of multiple sentences in each, we extend the notion of phrases so that they are independent of how information being communicated is split into sentences. Relations between the nodes of parse trees (which are other than syntactic) can merge phrases from different sentences or from a single sentence, which are not syntactically connected. We will refer to such extended phrases as thicket phrases.

We will consider two cases for text indexing, where establishing proper coreferences inside and between sentences connects entities in an index for proper match with a question:

Text for indexing 1: … *Tuberculosis is usually a lung disease. It is cured by doctors specializing in pulmonology.*

Text for indexing 2: … *Tuberculosis is a lung disease… Pulmonology specialist Jones was awarded a prize for curing a special form of disease.*

Question: *Which specialist doctor should treat my tuberculosis?*

In the first case, establishing coreference link *Tuberculosis* $\rightarrow$ *disease* $\rightarrow$ *is cured by doctors pulmonologists* helps to match these entities with the ones from the question. In the second case this portion of text does not serve as a relevant answer to the question, although it includes keywords from this question. Hence at indexing time, keywords should be chained not just by their occurrence in individual sentences, but additionally on the basis of coreferences. If words *X* and *Y* are connected by a coreference relation, an index needs to include the chain of words $X_0, X_1 \ldots X, Y_0, Y_1 \ldots Y$, where chains $X_0, X_1 \ldots X$ and $Y_0, Y_1 \ldots Y$ are already indexed (phrases including *X* and *Y*). Hence establishing coreferences is important to extend index in a way to improve search recall. Usually, keywords from different sentences can only be matched with query keywords with a low score (high score is delivered by inter-sentence match).

If we have two parse trees P_1 and P_2 of text T_1, and an arc for a relation $r: P_{1j} \rightarrow P_{2j}$ between the nodes P_{1j} and P_{2j}, we can now match $\ldots, P_{1i-2}, P_{1i-1}, P_{1i}, P_{2j}, P_{2j+1}, P_{2j+2}, \ldots$ of T_1 against a phrase of a single sentence or a merged phrases of multiple sentences from T_2.

2.1 Finding similarity between a question and an answer

We will compare the following approaches to assessing the similarity of questions and answers as paragraphs:

- Baseline: bag-of-words approach, which computes the set of common keywords/n-grams and their frequencies.
- Pair-wise matching: we will apply syntactic generalization to each pair of sentences, and sum up the resultant commonalities. This technique has been developed by Galitsky (2013).
- Paragraph-paragraph matching.

The first approach is most typical for industrial NLP applications today, and the second one was used in (Galitsky *et al.*, 2012). The kernel-based approach to parse tree similarities (Zhang *et al.*, 2008), as well as tree sequence kernel (Sun *et al.*, 2011), being tuned to parse trees of individual sentences, also belongs to the second approach.

We intend to demonstrate the richness of the approach being proposed, and in the consecutive sections we will provide a step-by-step explanation. We will introduce a pair of short texts (articles) and compare the above three approaches. The first paragraph can be viewed as a search query, and the second paragraph can be viewed as a candidate answer. A relevant answer should be a closely related text, which is not a piece of duplicate information.

"Iran refuses to accept the UN proposal to end the dispute over work on nuclear weapons",
"UN nuclear watchdog passes a resolution condemning Iran for developing a second uranium enrichment site in secret",
"A recent IAEA report presented diagrams that suggested Iran was secretly working on nuclear weapons",
"Iran envoy says its nuclear development is for peaceful purpose, and the material evidence against it has been fabricated by the US",
^
"UN passes a resolution condemning the work of Iran on nuclear weapons, in spite of Iran claims that its nuclear research is for peaceful purpose",
"Envoy of Iran to IAEA proceeds with the dispute over its nuclear program and develops an enrichment site in secret",
"Iran confirms that the evidence of its nuclear weapons program is fabricated by the US and proceeds with the second uranium enrichment site"

The list of common keywords gives a hint that both documents are on nuclear program of Iran, however it is hard to get more specific details.

Iran, UN, proposal, dispute, nuclear, weapons, passes, resolution, developing, enrichment, site, secret, condemning, second, uranium

Pair-wise generalization gives a more accurate account on what is common between these texts.

[NN-work IN-* IN-on JJ-nuclear NNS-weapons], [DT-the NN-dispute IN-over JJ-nuclear NNS-*], [VBZ-passes DT-a NN-resolution],
[VBG-condemning NNP-iran IN-*],
[VBG-developing DT-* NN-enrichment NN-site IN-in NN-secret]],
[DT-* JJ-second NN-uranium NN-enrichment NN-site]],
[VBZ-is IN-for JJ-peaceful NN-purpose],
[DT-the NN-evidence IN-* PRP-it], [VBN-* VBN-fabricated IN-by DT-the NNP-us]

Parse Thicket generalization gives the detailed similarity picture which looks more complete than the pair-wise sentence generalization result above. Please see also Fig. 3.

[NN-Iran VBG-developing DT-* NN-enrichment NN-site IN-in NN-secret]
[NN-generalization-<UN/nuclear watchdog> * VB-pass NN-resolution VBG condemning NN- Iran]
[NN-generalization-<Iran/envoy of Iran> Communicative_action DT-the NN-dispute IN-over JJ-nuclear NNS-*
[Communicative_action - NN-work IN-of NN-Iran IN-on JJ-nuclear NNS-weapons]
[NN-generalization <Iran/envoy to UN> Communicative_action NN-Iran NN-nuclear NN-* VBZ-is IN-for JJ-peaceful NN-purpose],
Communicative_action - NN-generalize <work/develop> IN-of NN-Iran IN-on JJ-nuclear NNS-weapons]*
[NN-generalization <Iran/envoy to UN> Communicative_action NN-evidence IN-against NN Iran NN-nuclear VBN-fabricated IN-by DT-the NNP-us]
condemn^proceed [enrichment site] <leads to> suggest^condemn [work Iran nuclear weapon]

"^" in the following example and through all the paper means *generalization* operation. Describing parse trees we use standard notation for constituency trees: **[…]** represents subphraze, **NN**, **JJ, NP** etc. denote parts-of-speech and types of subphrases, ***** is used to denote random tree node.

One can feel that PT-based generalization almost as complete as human would do in terms of similarity between texts. To obtain these results, we need to be capable of maintaining connections between sentences such as coreferences, and also of apply the relationships between entities to our analysis (entities, sub-entities, super-entities) obtained from WordNet or via web mining (Galitsky et al 2013). We also need to be able to identify communicative actions and generalize them together with their subjects according to the specific patterns of speech act theory, if a text describes an interaction between people. Moreover, we need to maintain rhetoric structure relationship between sentences, to generalize at a higher level above sentences irrespectively of how information is distributed through sentences. We define Parse Thicket as a set of parse trees for sentences with arcs for links between the words of different sentences. These arcs are for coreferences, entity-entity and rhetoric relations, and communicative actions.

The focus of this paper is to apply parse thickets and their generalization to search relevance where a query is a paragraph of text.

2.2 From phrase to paragraph-level generalization

Although the generalization is defined as the set of maximal common sub-graphs, its computation in this study is based on matching phrases. To generalize a pair of sentences, we perform chunking and extract all noun, verb, prepositional and other types of phrases from each sentence. Then we perform generalization for each type of phrases, attempting to find a maximal common sub-phrase for each pair of phrases of the same type. The resultant phrase-level generalization can then be interpreted as a set of paths in resultant common sub-trees (Galitsky *et al.*, 2012).

Thicket phrases are the regular phrases extended by the words from other sentences linked by inter-sentence arcs. The algorithm of forming thicket phrases is as follows. Most types of thicket arcs will be illustrated below. Please refer to (Galitsky *et al.*, 2012) for further details.

```
For each sentence $S$ in a paragraph $P$:
1  Form a list of previous sentences in a paragraph $S_{prev}$
2  For each word in the current sentence:
  2.1  If this word is a pronoun: find all nouns or noun phrases in the $S_{prev}$ which are:
         o  The same entities (via anaphora resolution)
  2.2  If this word is a noun: find all nouns or noun phrases in the $S_{prev}$ which are:
         o  The same entities (via anaphora resolution)
         o  Synonymous entity
         o  Super entities
         o  Sub and sibling entities
  2.3  If this word is a verb:
    2.3.1  If it is a communicative action:
      2.3.1.1 Form the phrase for its subject $VBCA_{phrase}$, including its verb phrase $VB_{phrase}$
      2.3.1.2 Find a preceding communicative action $VBCA_{phrase0}$ from $S_{prev}$ with its subject
      2.3.1.3 Form a thicket phrase [$VBCA_{phrase}$ , $VBCA_{phrase0}$]
    2.3.2  If it indicates RST relation:
      2.3.2.1 Form the phrase for the pair of phrases which are the subjects [VBRSTphrase1, VBRSTphrase2], of this RST relation, VBRSTphrase1 belongs to $S_{prev}$.
```

3 Arcs of parse thicket based on theories of discourse

We treat computationally the following approaches to textual discourse:

- Rhetoric structure theory (RST) (Mann *et al.*, 1992);
- Speech Act theory or shortly SpActT (Searle, 1969).

Although both these theories have psychological observation as foundations and are mostly of a non-computational nature, a specific computational framework need to be built for them (Galitsky *et al.*, 2010; 2013a). We use these sources to find links between sentences to enhance indexing for search. For RST, we attempt to extract an RST relation and form a thicket phrase around it, including a placeholder for RST relation itself. For SpActT, we use a vocabulary of *communicative actions* to find their subjects (Galitsky and Kuznetsov, 2008), add respective arcs to PT and form the respective set of thicket phrases.

RST example

Fig.1 shows the generalization instance based on RST relation “RCT-evidence” (Marcu, 1997). This relation occurs between the phrases *evidence-for-what [Iran’s nuclear weapon program] and what-happens-with-evidence [Fabricated by USA]* and *evidence-for-what [against Iran’s nuclear development]* and *what-happens-with-evidence [Fabricated by the USA].*

Notice that in the latter case we need to merge (perform anaphora substitution) the phrase *‘ its nuclear development’* with *‘evidence against it’* to obtain *‘evidence against its nuclear development’*. Notice the arc *it - development*, according to which this anaphora substitution occurred. *Evidence* is removed from the phrase because it is the indicator of RST relation, and we form the subject of this relation to match. Furthermore, we need another anaphora substitution *its* - *Iran* to obtain the final phrase.

As a result of generalizations of two RST relations of the same sort (evidence) we obtain *Iran nuclear NNP – RST-evidence – fabricate by USA.*

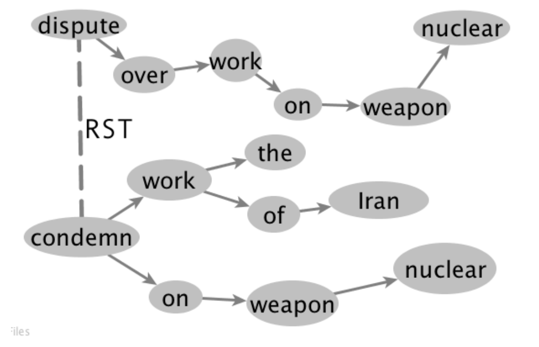

Fig.1: An example of the mapping for the rhetoric structures

Notice that we could not obtain this similarity expression by using sentence-level generalization.

Communicative actions example

Communicative actions are used by text authors to indicate the structure of a dialogue or a conflict (Searle, 1969). Hence analyzing the communicative actions' arcs of PT, one can find implicit similarities between texts. We can generalize:

- one communicative actions with its subject from T_1 against another communicative action with its subject from T_2 (communicative action arc is not used) ;
- a pair of communicative actions with their subjects from T_1 against another pair of communicative actions from T_2 (communicative action arcs are used).

In our example, we have the same communicative actions with subjects with low similarity:

condemn ['Iran for developing second enrichment site in secret'] vs condemn ['the work of Iran on nuclear weapon']

or different communicative actions with similar subjects.

The two distinct communicative actions dispute and condemn have rather similar subjects: 'work on nuclear weapon'. Generalizing two communicative actions with their subjects follows the rule: generalize communicative actions themselves, and 'attach' the result to generalization of their subjects as regular sub-tree generalization. Two communicative actions can always be generalized, which is not the case for their subjects: if their generalization result is empty, the generalization result of communicative actions with these subjects is empty too. The generalization result here for the case 1 above is:

condemn^dispute [work-Iran-on-nuclear-weapon].

Generalizing two different communicative actions is based on their attributes and is presented in (Galitsky *et al.*, 2013).

T_1 T_2

condemn [second uranium enrichment site] ↔ proceed [develop an enrichment site in secret]
↓ *communicative action arcs* ↓
suggest [Iran is secretly working on nuclear weapon] ↔ condemn [the work of Iran on nuclear weapon]

which results in *condemn^proceed [enrichment site] <leads to> suggest^condemn [work Iran nuclear weapon].*

Notice that generalization

condemn [second uranium enrichment site] ↔ condemn [the work of Iran on nuclear weapon]
↓ *communicative action arcs* ↓
suggest [Iran is secretly working on nuclear weapon] ↔ proceed [develop an enrichment site in secret]

gives zero result because the arguments of *condemn* from T_1 and T_2 are not very similar. Hence we generalize the subjects of communicative actions first before we generalize communicative actions themselves.

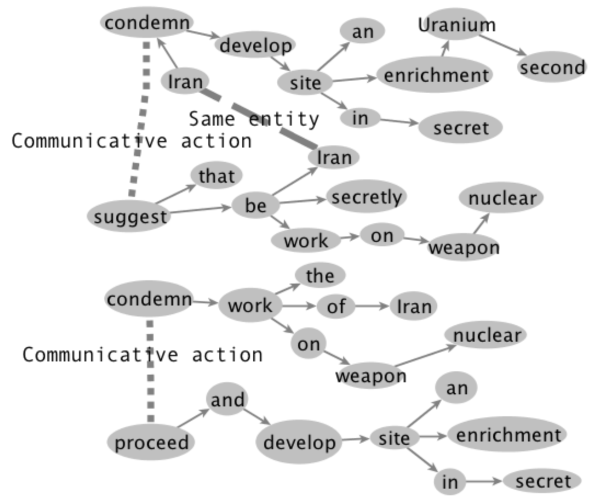

Fig. 2: A fragment of PT showing the mapping for the pairs of communicative actions.

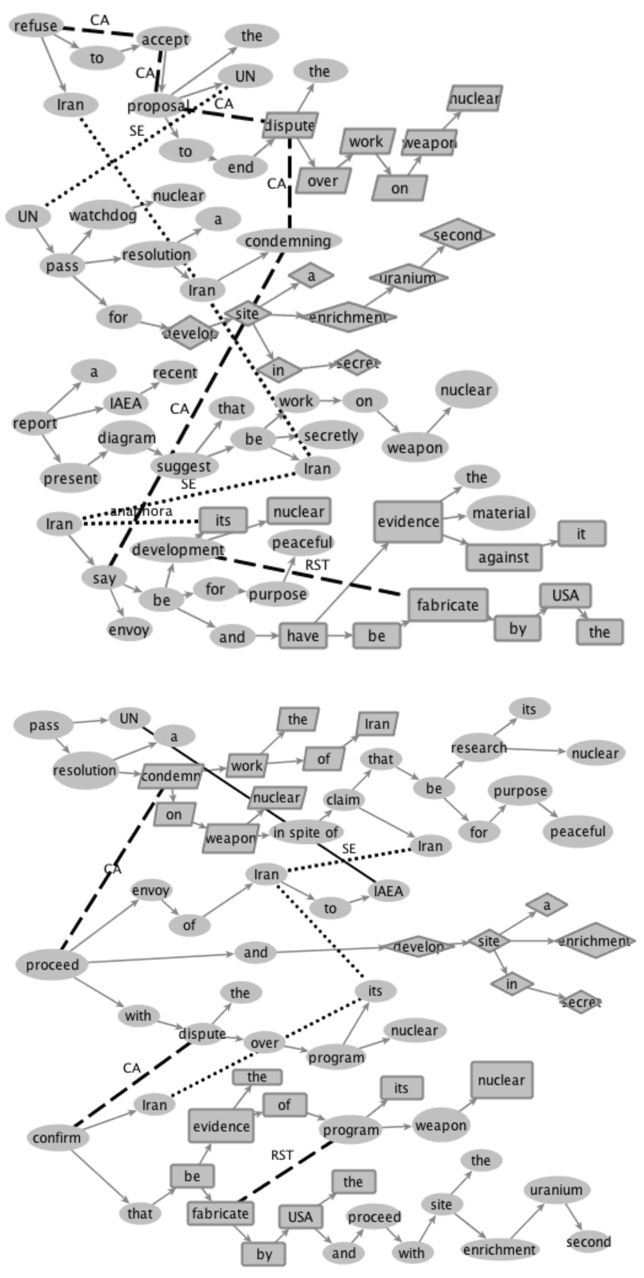

Fig.3: Finding similarity between two parse thickets. Groups of vertices with the same shape and dark-gray border show the maximum common sub-thickets, where the number of vertexes serves as a score for similarity between a question and answer.

3 Evaluation of multi-sentence question answering

We proceed to evaluation of how generalization of PTs can improve multi-sentence search, where one needs to compare a query as a paragraph of text against a candidate answer as a paragraph of text (search result snippet). Evaluation is based on a re-ranking search results obtained by Bing search engine API, relying on the PT similarity score. The similarity score is defined as a total number of vertexes in a common maximum subgraph. We approximate this estimate by calculating the number of words in maximal common sub-phrases, taking into account weight for parts of speech (Galitsky et al 2012).

Evaluation results are shown in Table 1. Three domains are used in evaluation:

- Product recommendation, where an agent reads chats about products and finds relevant information on the web about a particular product.
- Travel recommendation, where an agent reads chats about travel and finds relevant information on the travel websites about a hotel or an activity.
- Facebook recommendation, where an agent reads wall postings and chats, and finds a piece of relevant information for friends on the web.

In each of these domains we selected a portion of text on the web to form a query, and then filtered search results delivered by Bing search engine API. One can observe that unfiltered precision is 58.2%, whereas improvement by pair-wise sentence generalization is 11%, thicket phrases/snippets – additional 6%, and thicket phrases for original sentences in the documents – additional 1.5%.

One can also see that the higher the complexity of sentence, the higher the contribution of generalization technology, from sentence level to thicket phrases.

4 Algorithms and scalability of the approach

The generalization operation on parse trees for sentences and parse thickets for paragraphs is defined as finding a set of maximum common sub-trees and sub parse thickets respectively. Although for the trees this problem is *O(n)*, for the general case of graphs finding maximal common sub-graphs is NP-complete (Kann, 1992).

To estimate the complexity of generalization of two PT, let us consider an average case with five sentences in each paragraph and 15 words in each sentence. Such thickets have on average 10 phrases per sentence, 10 inter-sentence arcs, which give us up to 40 thicket phrases each.

Query type	Query complexity	Relevance of baseline Bing search, %,	Relevance single-sentence phrase-based generalization search, %	Relevance of thicket-based phrase generalization search, %, using snippets	Relevance of parse thicket-based phrase generalization search, %, using original sentences
Product recommendation search	1compound sent	62.3	69.1	72.4	72.9
	2 sent	61.5	70.5	71.9	72.8
	3 sent	59.9	66.2	72.0	73.4
	4 sent	60.4	66	68.5	69.2
Travel recommendation search	1compound	64.8	68	72.6	74.7
	2 sent	60.6	65.8	73.1	76.9
	3 sent	62.3	66.1	70.9	70.8
	4 sent	58.7	65.9	72.5	73.9
Facebo-ok friend agent support search	1compound	54.5	63.2	65.3	68.1
	2 sent	52.3	60.9	62.1	63.7
	3 sent	49.7	57	61.7	63.0
	4 sent	50.9	58.3	62.0	64.6
Avg		58.15	64.75	68.75	70.33

Table 1: Evaluation results

Hence for such parse thickets we have to generalize up to 50 linguistic phrases and 40 thicket phrases of the first thicket against the set of similar size for the second thicket. Taking into account a separate generalization of noun and verb phrases, this average case consists of 2* 45*45 generalizations, followed by the subsumption checks. Each phrase generalization is based on up to 12 string comparisons, taking an average size of phrase as 5 words. Hence on average the PT generalization includes 2*45*45*12*5 operations. Since a string comparison takes a few microseconds, thicket generalization takes on average 100 milliseconds without use of index. However, in an industrial search application where phrases are stored in an inverse index, the generalization operation can be completed in constant time, irrespectively of the size of index (Lin, 2013).

5 Conclusions

In this work we build the framework for generalizing PTs as sets of phrases to re-rank search results obtained via keyword search.

The operation of generalization to learn from parse trees for a pair of sentences turned out to be important for text relevance tasks. Once we extended it to learning parse thickets for two paragraphs, we observed that the relevance is further increased compared to the baseline (Bing search engine API), which relies on keyword statistics in the case of multi-sentence query.

We considered the following sources of relations between words in sentences: coreferences, taxonomic relations such as sub-entity, partial case, predicate for subject etc., rhetoric structure relation and speech acts. We demonstrated that search relevance can be improved if search results are subject to confirmation by parse thicket generalization, when answers occur in multiple sentences.

The system architecture serves as a basis of OpenNLP – similarity component, which is a separate Apache Software foundation project, accepting input from either OpenNLP or Stanford NLP. Code and libraries described here are available at http://code.google.com/p/relevance-based-on-parse-trees and http://svn.apache.org/repos/asf/opennlp/sandbox/opennlp-similarity/.

The system is ready to be plugged into Lucene library to improve search relevance. Also, a SOLR request handler is provided so that search engineers can switch to a PT-based multi-sentence search to quickly verify if relevance is improved. The system is designed for search engineers not familiar with linguistic technologies, who can plug in the richness of linguistic features of OpenNLP and Stanford NLP to work for them in a search application.

References

Galitsky, B. *Natural Language Question Answering System: Technique of Semantic Headers*. Advanced Knowledge International, Australia (2003).

Galitsky, B., Josep Lluis de la Rosa, Gábor Dobrocsi. *Inferring the semantic properties of sentences by mining syntactic parse trees*. Data & Knowledge Engineering. Volume 81-82, November (2012) 21-45.

Galitsky, B., Daniel Usikov, Sergei O. Kuznetsov: *Parse Thicket Representations for Answering Multi-sentence questions*. 20th International Conference on Conceptual Structures, ICCS 2013 (2013).

Galitsky, B., Kuznetsov S.O., *Learning communicative actions of conflicting human agents*. J. Exp. Theor. Artif. Intell. 20(4): 277-317 (2008).

Galitsky, B., Machine Learning of Syntactic Parse Trees for Search and Classification of Text. *Engineering Application of AI*, http://dx.doi.org/10.1016/j.engappai.2012.09.017, (2012).

Jiangning Wu, Zhaoguo Xuan and Donghua Pan, *Enhancing text representation for classification tasks with semantic graph structures*, International Journal of Innovative Computing, Information and Control (ICIC), Volume 7, Number 5(B).

Haussler, D. *Convolution kernels on discrete structures*, 1999.

Moschitti, A. *Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees*. In Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany, 2006.

Mann, William C., Christian M. I. M. Matthiessen and Sandra A. Thompson (1992). *Rhetorical Structure Theory and Text Analysis*. Discourse Description: Diverse linguistic analyses of a fund-raising text. ed. by W. C. Mann and S. A. Thompson. Amsterdam, John Benjamins: 39-78.

Searle, John. 1969. *Speech acts: An essay in the philosophy of language*. Cambridge, England: Cambridge University.

Sun, J., Min Zhang, Chew Lim Tan. *Tree Sequence Kernel for Natural Language*. AAAI-25, 2011.

Zhang, M.; Che, W.; Zhou, G.; Aw, A.; Tan, C.; Liu, T.; and Li, S. 2008. *Semantic role labeling using a grammar-driven convolution tree kernel*. IEEE transactions on audio, speech, and language processing 16(7):1315–1329.

Montaner, M.; Lopez, B.; de la Rosa, J. L. (June 2003). *A Taxonomy of Recommender Agents on the Internet*. Artificial Intelligence Review 19 (4): 285–330.

Collins, M., and Duffy, N. 2002. *Convolution kernels for natural language*. In Proceedings of NIPS, 625–632.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu and Dan Jurafsky. *Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational Linguistics* 39(4), 2013.

Plotkin, G.D. *A note on inductive generalization*. In B. Meltzer and D. Michie, editors, Machine Intelligence, volume 5, pages 153-163. Elsevier North-Holland, New York, 1970.

Lin, Jimmy. *Data-Intensive Text Processing with MapReduce*. intool.github.io/MapReduceAlgorithms/MapReduce-book-final.pdf , 2013.

Cascading en.wikipedia.org/wiki/Cascading. http://www.cascading.org/ 2013.

Dean, Jeff. *Challenges in Building Large-Scale Information Retrieval Systems*. research.google.com/people/jeff/WSDM09-keynote.pdf 2009.

Viggo Kann. 1992. *On the Approximability of the Maximum Common Subgraph Problem*. In (STACS '92), Alain Finkel and Matthias Jantzen (Eds.). Springer-Verlag, London, UK, UK, 377-388.

Lodhi, H.; Saunders, C.; Shawe-Taylor, J.; Cristianini, N.; and Watkins, C. 2002. *Text classification using string kernels*. The Journal of Machine Learning Research 2:419–444.

Moschitti, A. 2004. *A study on convolution kernels for shallow semantic parsing*. In Proceedings of ACL, 335–342.

Sun, J.; Zhang, M.; and Tan, C. 2010. *Exploring syntactic structural features for sub-tree alignment using bilingual tree kernels*. In Proceedings of ACL, 306–315.

Zhang, M.; Che, W.; Zhou, G.; Aw, A.; Tan, C.; Liu, T.; and Li, S. 2008. *Semantic role labeling using a grammar-driven convolution tree kernel. IEEE transactions on audio, speech,and language processing*. 16(7):1315–1329.

Zhang, M.; Zhou, G.; and Aw, A. 2008. *Exploring syntactic structured features over parse trees for relation extraction using kernel methods*. Information Processing & Management 44(2):687–701.

Byun, H. and Seong-Whan Lee. 2002. *Applications of Support Vector Machines for Pattern Recognition: A Survey*. In Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines (SVM '02), Seong-Whan Lee and Alessandro Verri (Eds.). Springer-Verlag, London, UK, UK, 213-236.

Manning, C. and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA: May 1999.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing. An Introduction to Natural*

Language Processing, Computational Linguistics, and Speech Recognition. 2008.

OpenNLP http://incubator.apache.org/opennlp/documentation/manual/opennlp.htm (2012).

Robinson J.A. *A machine-oriented logic based on the resolution principle*. Journal of the Association for Computing Machinery, 12:23-41, 1965.

Mill, J.S. (1843) *A system of logic, ratiocinative and inductive*. London.

Fukunaga, K. *Introduction to statistical pattern recognition* (2nd ed.), Academic Press Professional, Inc., San Diego, CA, 1990.

Finn, V.K. (1999) *On the synthesis of cognitive procedures and the problem of induction*. NTI Series 2, N1-2 pp. 8-45.

Mitchell, T. (1997) *Machine Learning*. McGraw Hill.

Furukawa, K. (1998) *From Deduction to Induction: Logical Perspective. The Logic Programming Paradigm*. In Apt, K.R., Marek V.W., Truszczynski, M., Warren, D.S., Eds. Springer.

Bharat Bhasker; K. Srikumar (2010). *Recommender Systems in E-Commerce*. CUP. ISBN 978-0-07-068067-8.

Hennig-Thurau, Thorsten, André Marchand, and Paul Marx. (2012), *Can Automated Group Recommender Systems Help Consumers Make Better Choices?* Journal of Marketing, 76 (5), 89-109.

Punyakanok, V.,Roth, D. and Yih, W. *The Necessity of Syntactic Parsing for Semantic Role Labeling*. IJCAI-05.

Domingos P. and Poon, H. *Unsupervised Semantic Parsing*, In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009. Singapore: ACL.

Marcu, D. (1997) *From Discourse Structures to Text Summaries*, in I. Mani and M.Maybury (eds) Proceedings of ACL Workshop on Intelligent Scalable Text Summarization, pp. 82–8, Madrid, Spain.

Abney, S. *Parsing by Chunks*, Principle-Based Parsing, Kluwer Academic Publishers, 1991, pp. 257-278.