ЧАЩА СИНТАКСИЧЕСКОГО РАЗБОРА ДЛЯ АБЗАЦА ТЕКСТА

Галицкий Б. (boris.galitsky@ebay.com),
Иворский Д. (dilv_ru@yahoo.com),
Кузнецов С. (skuznetsov@hse.ru),
Строк Ф. (fdr.strok@gmail.com)

eBay Inc.; Национальный исследовательский университет, Высшая школа экономики, Москва, Россия

Мы разрабатываем технику представления структуры предложений и абзацев текста в виде графов. Мы определяем чащу синтаксического разбора как объединение синтаксических деревьев разбора предложений. Чаща включает дуги между вершинами синтаксических деревьев для таких отношений, как кореферентность и таксономия. Эти дуги также получаются из других источников, в том числе, теории Риторических Структур и Речевых Актов. В работе предлагается алгоритм вычисления чащ разбора. Также в работе рассматриваются программные средства, предназначенные для построения чащ разбора и выполнения операции обобщения (пересечения) чащ разбора. На основе рассматриваемого подхода проводятся вычислительные эксперименты по улучшению поиска в случае, когда запрос представлен несколькими предложениями. Производится сравнение базового поиска, поиска с помощью сопоставления отдельных предложений и поиска с использованием Чащ разбора.

PARSE THICKET REPRESENTATIONS OF TEXT PARAGRAPHS

Galitsky B. (boris.galitsky@ebay.com),
Ivovsky D. (dilv_ru@yahoo.com),
Kuznetsov S. (skuznetsov@hse.ru),
Strok F. (fdr.strok@gmail.com)

eBay Inc.; National Research University Higher School of Economics, Moscow, Russia

We develop a graph representation and learning technique for parse structures for sentences and paragraphs of text. We introduce parse thicket as a set of syntactic parse trees augmented by a number of arcs for inter-sentence word-word relations such as coreference and taxonomies. These arcs are also derived from other sources, including Rhetoric Structure and Speech Act theory. We introduce respective indexing rules that identify inter-sentence relations and join phrases connected by these relations in the search index. We propose an algorithm for computing parse thickets from parse trees. We develop a framework for automatic building and generalizing of parse thickets. The proposed approach is used for evaluation in the product search where search queries include multiple sentences. We draw the comparison for search relevance improvement by pair-wise sentence generalization and thicket-level generalization.

Keywords: learning taxonomy, learning syntactic parse tree, syntactic generalization, search relevance, paragraph matching

1. Introduction

Parse trees have become a standard form of representing the linguistic structures of sentences. In this study we will attempt to represent a linguistic structure of a *text paragraph* 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 that includes parse trees for each sentence, as well as additional arcs for inter-sentence relationship between parse tree nodes for words.

In this paper we will define the operation of *generalization of text paragraphs* to assess similarity between portions of text. The use of generalization for similarity assessment is inspired by structured approaches to machine learning versus unstructured, statistical approaches where similarity is measured by a distance in feature space. Our intention is to extend the operation of the least general generalization (antiunification of logical formula)[14] towards structural representations of paragraph of texts. Hence we will define the operation of generalization on Parse Thickets and outline an algorithm for it.

This generalization operation is a base for number of text analysis applications such as search, classification, categorization, and content generation [9]. *Generalization of text paragraphs* is based on the operation of generalization of two sentences explored in our earlier studies [8]. In addition to learning generalizations of individual sentences, in this paper we study how the links between words in sentences other than syntactic ones can be used to compute similarity between texts. To compute generalization of a pair of paragraph, we performed a pair-wise generalization for each sentence in paragraphs. This approach ignores the richness of coreference information, and in the current study we develop graph-learning means, specifically oriented to represent paragraphs of text as respective PTs with nodes interconnected by arcs for a number of relations including coreference and taxonomy relations. We also consider such discourse-related theories as Rhetoric Structure (RST) [24] and Communicative Actions (CA) [23] as a source of arcs to augment PTs. These arcs will connect nodes for words both within and between parse trees for sentences.

It is significant to note that we used "out of the box" tools for constructing parse trees for sentences. For evaluation we used OpenNLP [16] and Stanford NLP [27] frameworks, which are intended for working with constituency-based trees. Also we used ETAP-3 system [20, 25, 26], built for working with dependency-based trees. This system was applied only to construct and visualize syntactic trees for sentences from basic examples. More details about programming components of our framework will be given in evaluation section.

2. Introducing Parse Thicket

Is it possible to find more commonalities between texts treating parse trees at a higher level? For that we need to extend the syntactic relations between the nodes of the syntactic dependency parse trees towards more general text discourse relations.

What relations can we add to the sets of parse trees to extend the match? 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. Relations between the nodes of parse trees that are other than syntactic can merge phrases from different sentences or from a single sentence which are not syntactically connected.

If we have two parse trees P_1 and P_2 of text T_1, and an arc for a relation r

r: $P_{1j} \rightarrow P_{2j}$ between the nodes P_{1j} and P_{2j}, we can match …,$P_{1,i-2}$, $P_{1, i-1}$, $P_{1, i}$, $P_{2,j}$, $P_{2,j+1}$, $P_{2,j+2}$, … of T_1 against a chunk of a single sentence of merged chunks of multiple sentences from T_2.

2.1. Finding similarity between two paragraphs of text

There are several approaches to assessing the similarity of text paragraphs:

- Baseline: bag-of-words approach, which computes the set of common keywords/ n-grams and their frequencies.
- Pair-wise matching: syntactic generalization to each pair of sentences, and summing up the resultant commonalities. This technique has been developed in our previous work [9].
- Paragraph-paragraph matching.

The first approach is most typical for industrial NLP applications today, and the second is the one of our previous studies. Kernel-based approach to parse tree similarities [13, 22], as well as tree sequence kernel [21], being tuned to parse trees of individual sentences, also belongs to the second approach.

Let us consider a short example to compare the above three approaches. Fragments of this example will be used through the whole paper. The generalization operation is denoted by '^':

"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",

$\wedge$

"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:

[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*]

One can feel that PT-based generalization closely approaches human performance in terms of finding similarities between texts. To obtain these results, we need to be capable of maintaining coreferences, apply the relationships between entities to our analysis *(subject vs relation-to-this subject),* including relationships between verbs *(develop* is a partial case of *work).* 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. Moreover, we need to maintain rhetoric structure relationships between sentences to generalize at a higher level above sentences.

The focus of this paper will be to introduce parse thicket and their generalization as paragraph-level structured representation. It will be done with the help of the above example. Fig. 1 and Fig. 2 show the dependency-based parse trees for the above texts T_1 and T_2. Each tree node has labels as part-of-speech and its form (such as SG for 'single'); also, tree edges are labeled with the syntactic connection type (such as 'composite'). Source trees were built and visualized using ETAP-3 system [20, 25, 26]. Then we added specific "red" arcs to them in order to illustrate the idea of parse thicket.

Generalization of parse thickets, being the set of maximal common sub-graph (sub-parse thicket) can be computed at the level of phrases as well, as the set of maximal common sub-phrases. However, the notion of phrases is extended now: *thicket phrases* can contain regular phrases from different sentences. The way these phrases are extracted and formed depends on the source of non-syntactic link between words in different sentences: thicket phrases are formed in a different way for communicative actions and RST relations.

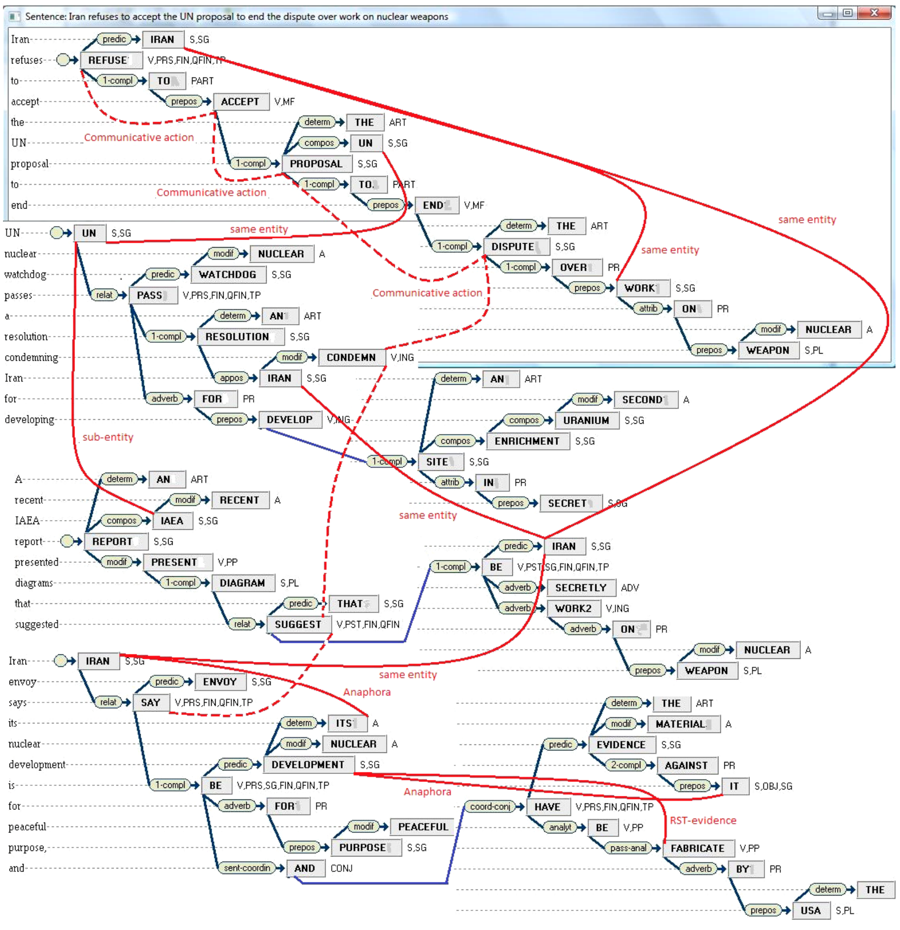
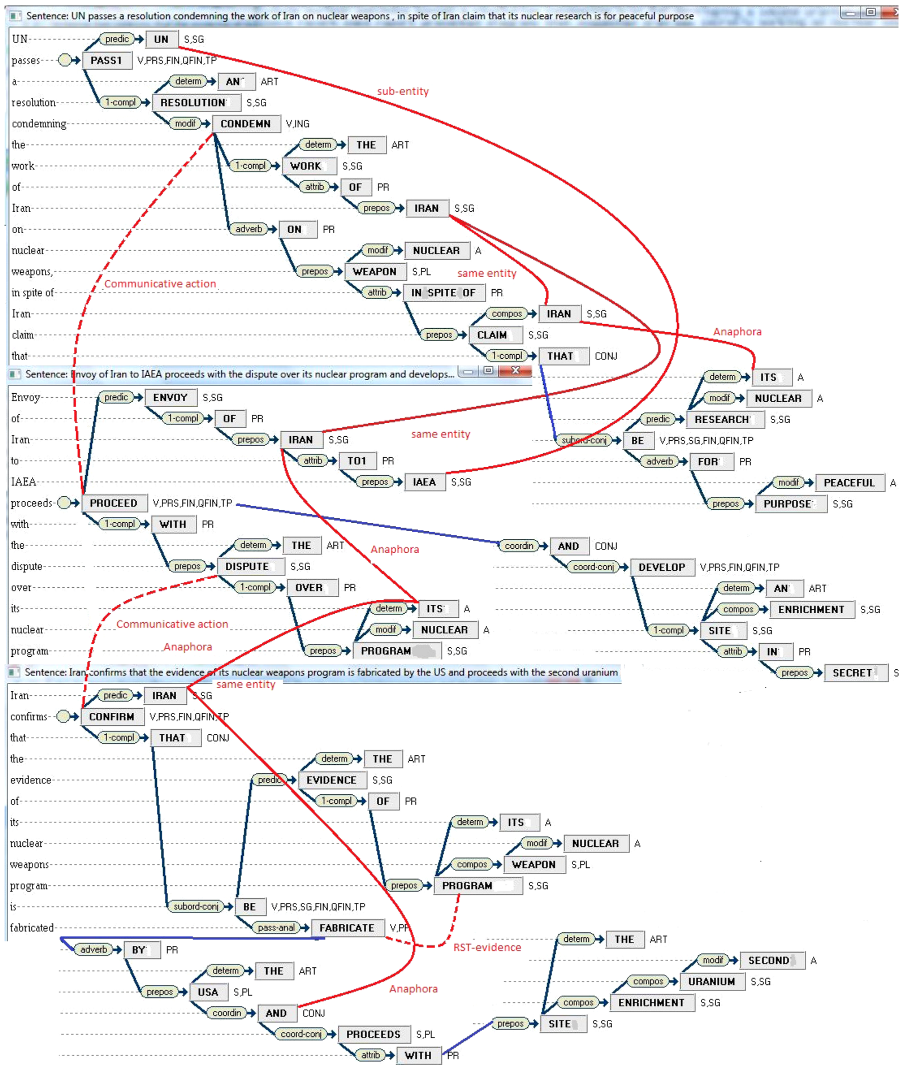


Fig. 1: Parse thicket for text T_1


Sentence: UN passes a resolution condemning the work of Iran on nuclear weapons , in spite of Iran claim that its nuclear research is for peaceful purpose

Sentence: Envoy of Iran to IAEA proceeds with the dispute over its nuclear program and develops...

Sentence: Iran confirms that the evidence of its nuclear weapons program is fabricated by the US and proceeds with the second uranium

Fig. 2: Parse thicket for text T_2

3. Arcs of parse thicket based on theories of discourse

Using the unified framework we develop two approaches to textual discourse based on

- Rhetoric structure theory (RST) [24],
- Communicative Actions (CA) [23].

We used a vocabulary of Communicative actions to

1. find their subjects [23],
2. add respective arcs to the parse thicket,
3. index combination of phrases as subjects of communicative actions

For RST, we introduce explicit indexing rules which will be applied to each paragraph and

1. attempt to extract an RST relation,
2. build corresponding fragment of the parse thicket, and
3. index respective combination of formed phrases (noun, verb, prepositional), including words from different sentences.

3.1. Generalization based on Rhetoric structure arcs

The theory of Rhetoric structures (RST)[24] was developed to explain the coherence of texts, seen as a kind of function, linking parts of a text to each other.

Two connected clouds represented on the right of Fig.3 show the generalization instance based on RST relation "RCT-evidence". This relation occurs between the phrases

evidence-for-what [Iran's nuclear weapon program] and *what-happens-with-evidence [Fabricated by USA]* on the right-bottom, and

evidence-for-what [against Iran's nuclear development] and *what-happens-with-evidence [Fabricated by the USA]* on the right-top.

Notice that in the latter case we need to merge (perform anaphora substitution) the phrases '*its nuclear development*' and '*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.

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

Green clouds indicate the sub-PTs of T_1 and T_2, which are matched. We show three instances of PT generalization.

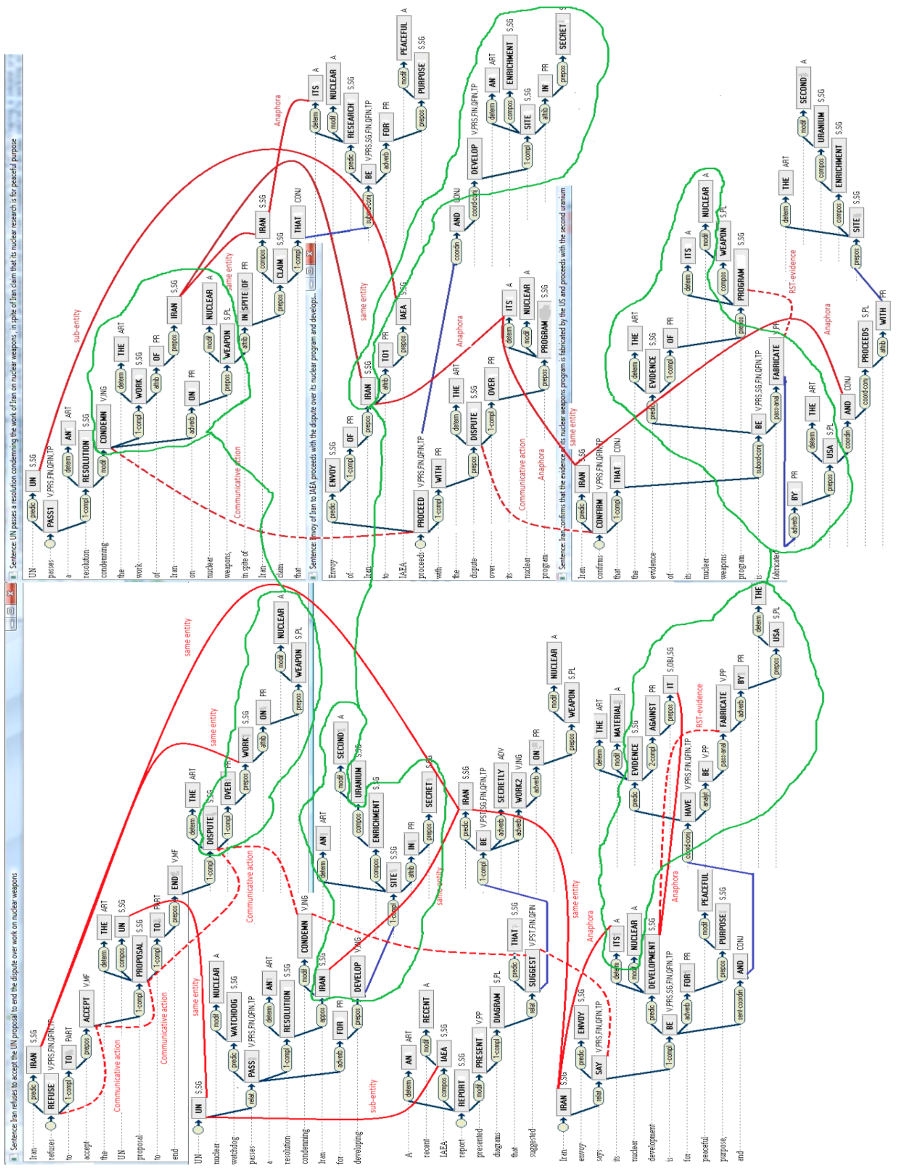


Fig. 3: Three instances of matching between sub-PTs shown as connected clouds

3.2. Generalization based on Communicative action arcs

Communicative actions (CA) are used by text authors to indicate a structure of a dialogue or a conflict [23]. Hence, analyzing the arcs of communicative actions of PT, one can find implicit similarities between texts. We can generalize

1. one communicative action with its subject from T_1 against another communicative action with its subject from T_2 (communicative action arc is not used) ;
2. 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.

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 elsewhere [7].

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

$$\begin{tikzcd} \textit{condemn [second uranium enrichment site]} \arrow[r, leftrightarrow] \arrow[d, "\text{communicative action arcs}"] & \textit{condemn [the work of Iran on nuclear weapon]} \arrow[d] \\ \textit{suggest [Iran is secretly working on nuclear weapon]} \arrow[r, leftrightarrow] & \textit{proceed [develop an enrichment site in secret]} \end{tikzcd}$$

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.

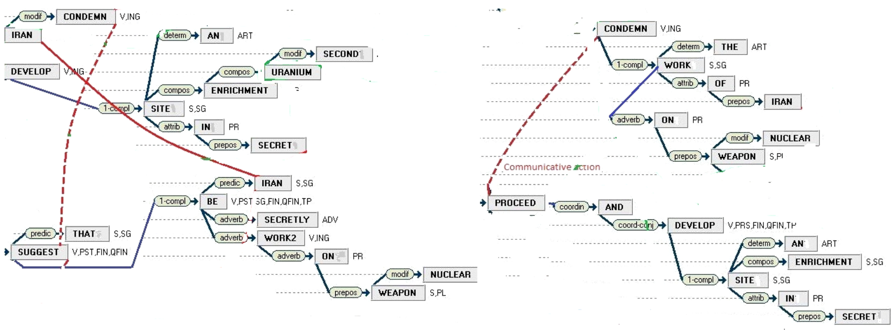


Fig. 4: A fragment of PT showing the mapping for the pairs of communicative actions

4. Generalization of thickets

4.1. Definition of generalization operation on two parse thickets

Given two parse thickets $C_x=(V_x, E_x)$ and $C_y=(V_y, E_y)$, their generalization denoted by $C_x \wedge C_y$ is defined as the set $\{G_1, G_2,\ldots,G_k\}$ of all inclusion-maximal common subgraphs of C_x and C_y. See e.g. [28].

4.2. Algorithm for forming thicket phrases for generalization

We will now outline the algorithm of forming thicket phrases.

```
For each sentence $S$ in a paragraph $P$
    Form a list of previous sentences in a paragraph $S_{prev}$
    For each word in the current sentence:
        - If this word is a *pronoun*: find all nouns or noun phrases in the $S_{prev}$,
          which are
            * The same entities (via anaphora resolution)
        - If this word is a *noun*: find all nouns or noun phrases in the $S_{prev}$, which are
            * The same entities (via anaphora resolution)
            * Synonymous entity
            * Super entities
            * Sub and sibling entities
        - If this word is a *verb*:
            * If it is a communicative action:
                Form the phrase for its subject $VBCA_{phrase}$, including its verb
                    phrase $VB_{phrase}$
                Find a preceding communicative action $VBCA_{phrase0}$
                    from $S_{prev}$ with its subject and form a thicket phrase
                    [$VBCA_{phrase}$, $VBCA_{phrase0}$]
            * If it indicates RST relation
                Form the phrase for the pair of phrases, which are the subjects
                [$VBRST_{phrase1}$, $VBRST_{phrase2}$], of this RST relation, $VBRST_{phrase1}$
                belongs to $S_{prev}$.
```

4.3. Sentence-level generalization algorithm

Below we outline the algorithm on finding a maximal sub-phrase for a pair of phrases, applied to the sets of thicket phrases for T_1 and T_2.

1. Split parse trees for sentences into sub-trees which are phrases for each type: *verb, noun, prepositional* and others; these sub-trees are overlapping. The sub-trees are coded so that information about occurrence in the full tree is retained.
2. All sub-trees are grouped by phrase types.
3. Extending the list of phrases by adding equivalence transformations
4. Generalize each pair of sub-trees for both sentences for each phrase type.
5. For each pair of sub-trees yield an alignment, and then generalize each node for this alignment. For the obtained set of trees (generalization results), calculate the score.
6. For each pair of sub-trees for phrases, select the set of generalizations with highest score (least general).
7. Form the sets of generalizations for each phrase types whose elements are sets of generalizations for this type.
8. Filtering the list of generalization results: for the list of generalization for each phrase type, exclude more general elements from lists of generalization for given pair of phrases.

5. Evaluation of multi-sentence search

5.1. PT-processing framework

There are two system components, which include Parse Thicket building and phrase-level processing.

The textual input is subject to a conventional text processing flow such as sentence splitting, tokenizing, stemming, part-of-speech assignment, building of parse trees and coreferences assignment for each sentence. Unlike our previous studies [19] computing parse thickets becomes fully automatic and includes not only RST and CA arcs but also coreference (via anaphora resolution) and taxonomic (same-entity, sub-entity, super-entity) relations. This flow is implemented by either OpenNLP or Stanford NLP, and the parse thicket is built based on the algorithm presented in this paper. The coreferences and RST component strongly relies on Stanford NLP's rule-based approach to finding correlated mentions based on the multi-pass sieves.

Phrase-level processing for the phrases of individual sentences has been described in detail in our previous studies [8, 19]. In this study we collect all phrases for all sentences of one paragraph of text, augment them with thicket phrases (linguistic phrases which are merged based on the inter-sentence relation), and generalize against that of the other paragraph of text.

5.2. Evaluation results

Having described the system architecture and engineering aspects, 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 (snippet). We refer the reader to [8] for the details on evaluation settings.

Table 1: Evaluation results

Query type	Query complexity	Relevance of baseline Bing search, %, averaging over 100 searches	Relevance single-sentence phrase-based generalization search, %, averaging over 100 searches	Relevance of thicket-based phrase generalization search, %, averaging over 100 searches
Product recommendation search	1 compound sentence	62.30	69.10	72.40
	2 sentences	61.50	70.50	71.90
	3 sentences	59.90	66.20	72.00
	4 sentences	60.40	66.00	68.50
Travel recommendation search	1 compound sentence	64.80	68.00	72.60
	2 sentences	60.60	65.80	73.10
	3 sentences	62.30	66.10	70.90
	4 sentences	58.70	65.90	72.50
Facebook friend agent support search	1compound sentence	54.50	63.20	65.30
	2 sentences	52.30	60.90	62.10
	3 sentences	49.70	57.00	61.70
	4 sentences	50.90	58.30	62.00
Average		58.15	64.75	68.75

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 give additional 6%. One can also see that the higher the complexity of sentence, the higher the contribution of generalization technology, from sentence level to thicket phrases.

6. Related work and conclusions

Usually, classical approaches to semantic inference rely on complex logical representations. However, practical applications usually adopt shallower lexical or lexical-syntactic representations, but lack a principled inference framework. Paper [2] proposed a generic semantic inference framework that operates directly on syntactic trees. New trees are inferred by applying entailment rules, which provide a unified representation for varying types of inferences. The current work deals with syntactic tree transformation in the graph-learning framework, treating various phrasings for the same meaning in a more unified and automated manner.

In our previous works we observed how employing a richer set of linguistic information such as syntactic relations between words assists relevance tasks [8, 9, 19]. To take advantage of semantic discourse information, we introduced parse thicket representation and proposed the way to compute similarity between texts based on generalization of parse thickets. In this work we build the framework for generalizing PTs as sets of phrases.

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. Parse thicket is intended to represent the syntactic structure of text as well as a number of semantic relations for indexing. To this end, a parse thicket contains relations between words in different sentences, such that these relations are essential to match queries with portions of texts to serve as answers.

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.

Using semantic information for query ranking has been proposed in [1]. Moreover, relying on matching of parse trees of a question and an answer has been the subject of [13] and [15]. However we believe that our study improves multi-sentence search, relying both on learning with parse tickets as connected parse trees and on linguistic theories on text coherence.

References

1. *Aleman-Meza, B., Halaschek, C., Arpinar, I. and Sheth, A.* “A Context-Aware Semantic Association Ranking,” Proc. First Int’l Workshop Semantic Web and Databases (SWDB ‘03), pp. 33–50, 2003.
2. *Bar-Haim, R., Dagan, I., Greental, I. Shnarch, E.* Semantic Inference at the Lexical-Syntactic Level AAAI-05.
3. *Bhogal, J., Macfarlane, A., Smith, P.* A review of ontology based query expansion. Information Processing & Management. Volume 43, Issue 4, July 2007, Pages 866–886.
4. *Sadid, C. Y., Hasan, A., Joty, S. R.:* Improving graph-based random walks for complex question answering using syntactic, shallow semantic and extended string subsequence kernels. Inf. Process. Manage. 47(6): 843–855 (2011).
5. *Ercan, G., Cicekli, I.* Using lexical chains for keyword extraction. Information Processing & Management, Volume 43, Issue 6, November 2007, Pages 1705–1714.
6. *Galitsky, B.* Natural Language Question Answering System. Technique of Semantic Headers. Advanced Knowledge International, Australia (2003).
7. *Galitsky, B., González, M. P., Chesñevar, C. I.* A novel approach for classifying customer complaints through graphs similarities in argumentative dialogue. Decision Support Systems, 46–3, 717–729 (2009).
8. *Galitsky, B., de la Rosa, J. L., Dobrocsi, G.* Inferring the semantic properties of sentences by mining syntactic parse trees. Data & Knowledge Engineering. Volume 81–82, November (2012a) 21–45.
9. *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, (2012b).
10. *Kapoor, S., Ramesh, H.* Algorithms for Enumerating All Spanning Trees of Undirected and Weighted Graphs, SIAM J. Computing, vol. 24, pp. 247–265, 1995.
11. *Jung-Jae, K., Pezik, P.* and Rebholz-Schuhmann, D. MedEvi: Retrieving textual evidence of relations between biomedical concepts from Medline. Bioinformatics. Volume 24, Issue 11 pp. 1410–1412 (2008).
12. *Mann, W. C., Matthiessen C. and Thompson, S.* (1992). Rhetorical Structure Theory and Text Analysis. Ed. by W. C. Mann and S. A. Thompson. Amsterdam, John Benjamins: 39–78.
13. *Moschitti, A.* Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany, 2006.
14. *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.
15. *Punyakanok, V., Roth, D., & Yih, W.* (2004). Mapping dependencies trees: an application to question answering. In: Proceedings of AI & Math, Florida, USA.
16. *OpenNLP* 2013. http://incubator.apache.org/opennlp/documentation/manual/opennlp.htm

17. *Marcu, D.* 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, 1997.
18. *Mihalcea and Tarau.* TextRank: Bringing Order into Texts. Empirial Methods in NLP 2004.
19. *Galitsky, B., Usikov, D., Kuznetsov, S. O.:* Parse Thicket Representations for Answering Multi-sentence questions. 20th International Conference on Conceptual Structures, ICCS 2013 (2013).
20. *Apresian, J., Boguslavsky, I., Iomdin, L., Lazursky, A., Sannikov, V., Sizov, V., Tsinman, L.* ETAP-3 linguistic processor: A full-fledged NLP implementation of the MTT //MTT 2003, First International Conference on Meaning — Text Theory. — 2003. — C. 279–288.
21. *Sun, J., Zhang, M., Lim Tan, C.* Tree Sequence Kernel for Natural Language. AAAI-25, 2011.
22. *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.
23. *Searle, J.* Speech acts: An essay in the philosophy of language. Cambridge, England: Cambridge University. 1969.
24. *Galitsky, B., Kuznetsov S. O.* Learning communicative actions of conflicting human agents. J. Exp. Theor. Artif. Intell. 20(4): 277–317 (2008).
25. *Apresian, J., Boguslavsky, I., Iomdin, L., Tsinman, L.* Lexical Functions as a Tool of ETAP-3. First International Conference on Meaning-Text Theory (MTT'2003). June 16–18, 2003. Paris: École Normale Supérieure, 2003.
26. *Иомдин Л., Петроченков В., Сизов В., Цинман Л.* Синтаксический анализатор системы Этап и его современное состояние. Papers from the national annual conference "Dialogue", 2012.
27. *Stanford NLP.* http://nlp.stanford.edu/
28. *Kuznetsov, S. O. and Samokhin, M. V.* Learning Closed Sets of Labeled Graphs for Chemical Applications. In: Proc. 15th Conference on Inductive Logic Programming (ILP 2005), Lecture Notes in Artificial Intelligence (Springer), Vol. 3625, pp. 190–208., 2005.