Learning Sub-structures of Document Semantic Graphs for Document Summarization

Jure Leskovec
Jozef Stefan Institute
Ljubljana, Slovenia
jure.leskovec@ijs.si

Marko Grobelnik
Jozef Stefan Institute
Ljubljana, Slovenia
marko.grobelnik@ijs.si

Natasa Milic-Frayling
Microsoft Research
Cambridge, UK
natasamf@microsoft.com

ABSTRACT
In this paper we present a method for summarizing document by creating a semantic graph of the original document and identifying the substructure of such a graph that can be used to extract sentences for a document summary. We start with deep syntactic analysis of the text and, for each sentence, extract logical form triples, subject–predicate–object. We then apply cross-sentence pronoun resolution, co-reference resolution, and semantic normalization to refine the set of triples and merge them into a semantic graph. This procedure is applied to both documents and corresponding summary extracts. We train linear Support Vector Machine on the logical form triples to learn how to extract triples that belong to sentences in document summaries. The classifier is then used for automatic creation of document summaries of test data. Our experiments with the DUC 2002 data show that increasing the set of attributes to include semantic properties and topological graph properties of logical triples yields statistically significant improvement of the micro-average F1 measure for the extracted summaries. We also observe that attributes describing various aspects of semantic graph are weighted highly by SVM in the learned model.

Categories and Subject Descriptors
H.3.1 [**Content Analysis and Indexing**]: Abstracting Methods – *document extracts, semantic structure, deep syntactic analysis.*

General Terms
Performance, Experimentation.

Keywords
Summarization, sentence extraction, abstract, document summary, semantic structure, linguistic analysis, machine learning, support vector machines.

LinkKDD 2004, August 2004, Seattle, WA, USA.

1. INTRODUCTION
Document summarization refers to the task of creating document surrogates that are smaller in size but retain various characteristics of the original document, depending on the intended use. While abstracts created by authors or trained professionals involve rewriting of text, automatic summarization of documents has been focused on extracting sentences from text [5, 10, 13, 15] so that the overall summary satisfies various criteria: optimal reduction of text for use in text indexing and retrieval [3], coverage of document themes [14], and similar. The objective of automatically creating professional abstracts of documents has been also pursued. In particular, the Document Understanding Conference [4] provides experimentation framework and a forum for exchanging research ideas and results on that particular topic.

Automated summarization is often approached in two phases. First, key textual elements, e.g., keywords, concepts, and concept relations, are extracted from the text using linguistic and statistical analysis. These are then used to select sentences from the text, enforcing various requirements on coverage and coherence of extracts [15, 16]. More sophisticated approaches involve generation of text based on textual units identified in the first phase.

In this paper we are primarily concerned with the first phase, i.e., identification of textual elements for use in extracting summaries. We start from the assumption that capturing semantic structure of a document is essential for summarization. We thus create semantic representations of the document, visualized as semantic graphs, and learn the model to extract sub-structures that could be used in document extracts. The basic elements of our semantic graphs are logical form triples, subject–predicate–object. We characterize each triple by a rich set of linguistic, statistical, and graph attributes and train linear SVM classifier to identify those triples that could be used for sentence extraction. Our experiments show that characteristics of the semantic graphs, obtained through sophisticated linguistic analyses, are most prominent attributes in the learnt SVM model. We also show that including the rich set of linguistic attributes increased the performance of the summarization model.

In the following sections we describe the procedure that we use for generating semantic graphs. We then discuss the experiment set up and the results of the sub-graph learning experiments and conclude by outlining the future work. We refer to background research and related work.

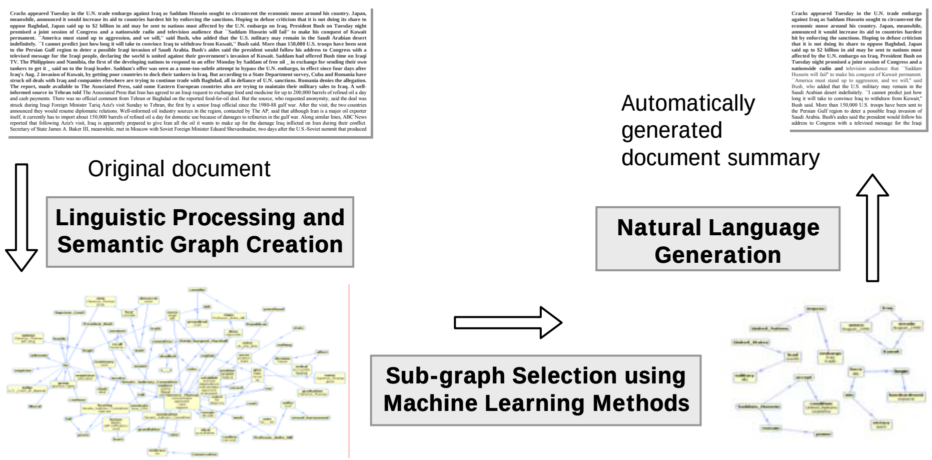

Figure 1. Summarization procedure based on semantic structure analysis.

2. PROBLEM STATEMENT

The task of automated document summarization is to produce a shorter version of an original document. As document summaries, in form of abstracts, have been generated by authors and trained professionals, it seems most natural to try to model human abstracting. However, evaluation of such models is difficult because of the text generation aspects. In this study we thus focus on learning models for extracting rather than generating summaries from document text.

We apply machine learning algorithms to capture characteristics of human extracted summary sentences. In contrast to related studies, which typically rely on a minimal understanding of the semantic structure of documents [5, 10], we start with deep syntactic analysis of the text. We extract elementary syntactic structures from individual sentences in the form of logical form triples, i.e., subject–predicate–object triples, and use semantic properties of nodes in the triples to build semantic graphs for both documents and corresponding summaries. We expect that extracted summaries would capture essential semantic relations within the document and thus their structures could be found within the document semantic graphs. We reduce the problem of summarization to acquiring machine learning models for mapping between the document graph and the graph of a summary. This means we learn models for extracting sub-structures from document semantic graphs which are characteristic of human selected summaries. We use logical form triples as basic features and apply Support Vector Machines to learn the summarization model.

3. SEMANTIC GRAPH GENERATION

Our approach to generating semantic representations of documents and summaries involves five phases, each described in more details in the sections below:

- Deep syntactic analysis – We apply deep syntactic analysis to document sentences, using NLPWin linguistic tool [3], and extract logical form triples
- Co-reference resolution – We identify co-references for named entities through surface form matching and text layout analysis, thus aiming at consolidating expressions that refer to the same named entity
- Pronomial reference resolution – We then use NLPWin syntactic and semantic tags to trace and resolve pronominal references as they appear in the text
- Semantic normalization – We expand terms in the logical form triples using WordNet semantic network in order to normalize expressions that refer to the same concepts
- Semantic graph analysis – We merge the logical form triples into a semantic graph and analyze the graph properties.

3.1 Linguistic Analysis

For linguistic analysis of text we use Microsoft's NLPWin natural language processing tool. NLPWin first segments the text into individual sentences, converts sentence text into a parse tree that represents the syntactic structure of the text (Figure 2) and then produces a sentence logical form that reflects the meaning, i.e., semantic structure of the text (Figure 3). This process involves a variety of techniques: use of knowledge base, grammar rules, and probabilistic methods in analyzing the text.

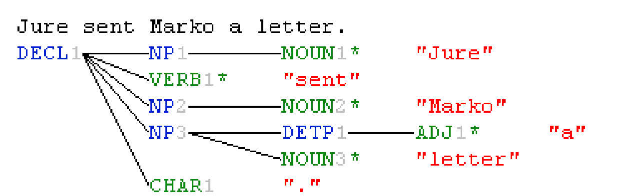

Figure 2. Syntactic tree for the sentence "Jure sent Marko a letter"

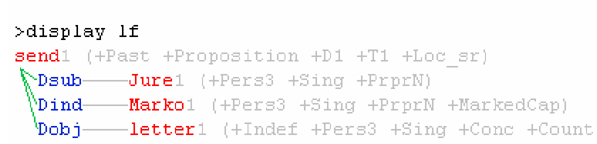

Figure 3. Logical form for the sentence

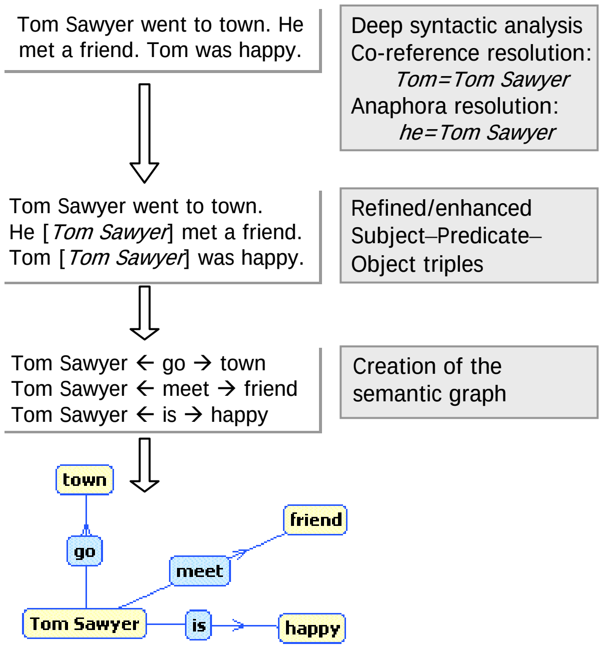

Figure 4. Process of creating a semantic graph.

The logical form for the sentence, shown in Figure 3, shows that the sentence is about sending, where "Jure" is the deep subject (an "Agent" of the activity), "Marko" is the deep indirect object (having a "Benefactive" role), and the "letter" is the deep direct object (assuming the "Patient" role). The notations in parentheses provide Semantic information about each node in the graphs. For example, "Jure" is a masculine, singular, and proper name.

From the logical form we extract constituent sub-structures in the form of triples: "Jure"←"send"→"Marko" and "Jure"←"send" →"letter". For each node we preserve semantic tags that are assigned by the NLPWin software. These are used in our further linguistic analyses and machine learning stage.

Figure 4 shows an example that outlines the main processes and output of different stages involved in generating the semantic graph. We first perform deep syntactic analysis to the raw text, using NLPWin software. Then we refine the analysis applying several methods: co-reference resolution, pronominal anaphora resolution, and semantic normalization (discussed in Section 3.3). Resulting triples are then linked based on common concepts into a semantic graph. Figure 6 shows an example of a semantic graph for an entire document.

3.2 Co-reference Resolution For Named Entities

In documents it is common that terms with different surface forms refer to the same entity. Identifying such terms is referred to as co-reference resolution. We restrict our co-reference resolution attempt to syntactic nodes that, in the NLPWin analysis, have the attribute of 'named entity'. Such are names of people, places, companies, and similar.

For each named entity we record the gender tag that was obtained based on NLPWin semantic resources. This narrows the set of named entity surface forms we compare with the new record. Then, starting with multi-word named entities, we first eliminate the standard set of English stop words and 'common' words, such as "Mr.", "Mrs.", "international", "company", "group", "president", "federal", etc. We then apply a simple rule by which two terms with distinct surface forms refer to the same entity if all the words from one term also appear as words in the other term. The algorithm, for example, correctly finds that "Hillary Rodham Clinton", "Hillary Clinton", "Hillary Rodham", and "Mrs. Clinton" all refer to the same entity.

3.3 Pronomial Anaphora Resolution

NLPWin automatically performs pronoun reference resolution within a single sentence. However, as NLPWin analysis is focused on individual sentences, those pronominal references that cross the sentence boundaries have to be resolved as a post process to the logical form analysis.

We take the approach that is similar to Mitkov in [11]. We start with resolved co-references to named-entities and make a simplifying assumption that pronouns in the text can refer only to named entities mentioned in the same text. We, accordingly, focus on resolving only five basic types of pronouns including all different forms: "he" ("his", "him", "himself"), "she", "I", "they" and "who". For a given pronoun, we search sequentially through sentences, backward and forward, and check the type of named entity in order to find suitable candidates. To find candidates we first search backward inside the sentence, then we continue searching backward in previous sentences. If no appropriate candidates are found so far, we extend our search in forward direction – first inside the sentence of the pronoun and then also inside the sentences that follow it. Each candidate is scored based on character and sentence distance and the direction from the pronoun, part of speech attributes, and other linguistic features. Among the scored candidate entities we chose the one with the best score and assign it to the pronoun.

Table 1: Performance of anaphora resolution algorithm

Pronoun	Frequency	Frequency [%]	Accuracy [%]
He	681	45.22	86.9
They	244	16.20	67.2
It	204	13.55	
I	64	4.25	82.8
You	50	3.32	
We	44	2.92	
That	44	2.92	
What	27	1.79	
She	24	1.59	62.5
This	22	1.46	
Who	11	0.73	63.6
Total	1506		82.1

We evaluated our pronoun resolution algorithm on 91 manually labeled documents from DUC 2002 dataset. The set contained 1506 pronouns of which 1024 (68%) were from the above

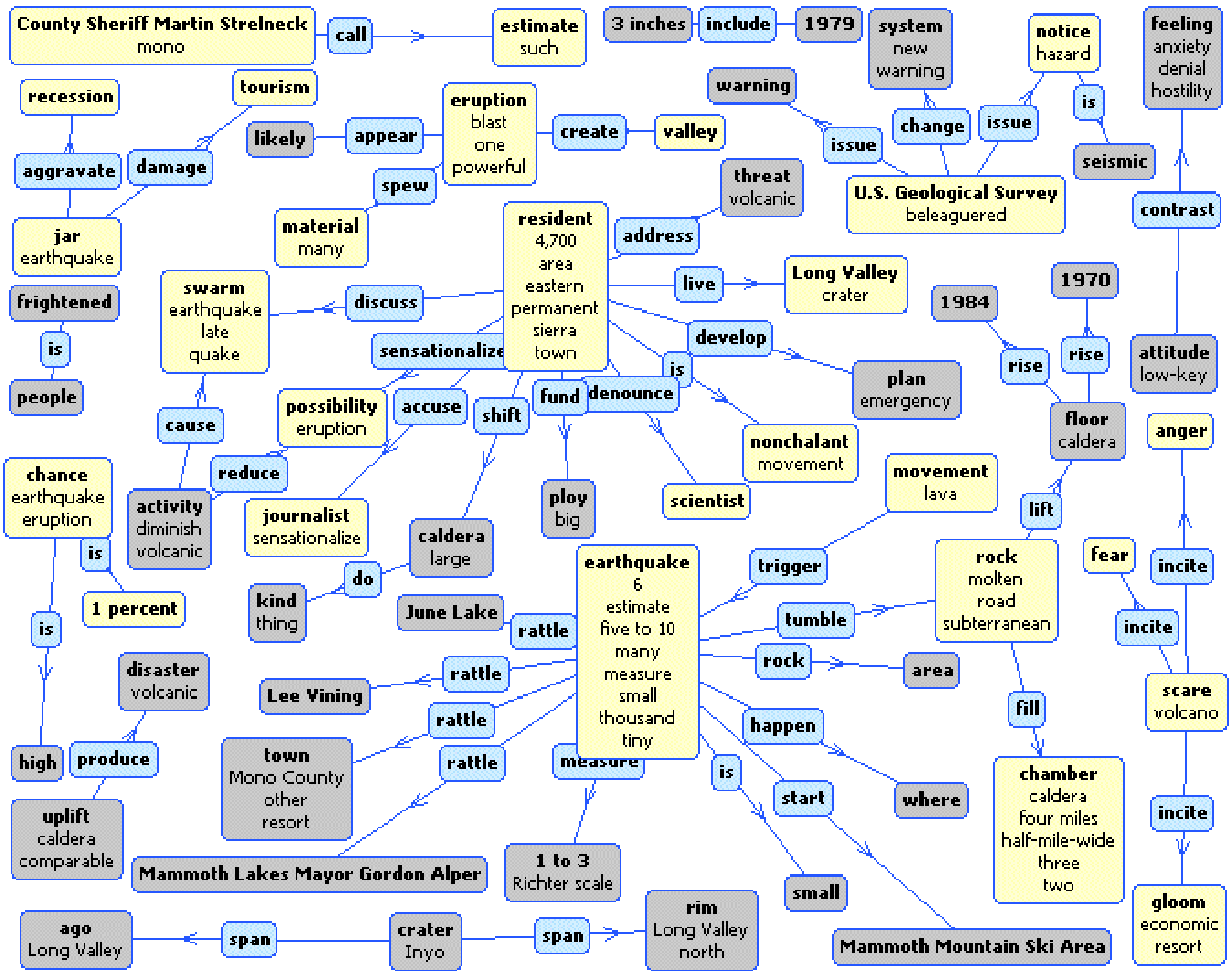

Figure **5: Full semantic graph of the document "Long Valley volcano activities". Subject/object nodes indicated by the light color (yellow) nodes in the graph indicate summary nodes. Gray nodes indicate non-summary nodes. We learn a model for distinguishing between the light and dark nodes in the graph.**

selected pronoun types and other 432 (32%) were "it", "you", "we", "what", and similar. Table 1 shows the distribution of most frequent pronouns and the accuracy of resolving the pronouns using our approach. Average accuracy over the 5 selected pronoun types is 82.1%. Scoring function coefficients were optimized through cross-validation experiments on the same data set.

3.4 Semantic Normalization

Having completed the pronoun and co-reference resolution, we arrive at a list of triples extracted from individual sentences and would like to link them into a semantic graph. In order to increase the coherence and compactness of the graphs, we decided to use WordNet [6] to establish synonymous relationships among nodes.

WordNet is a semantic network of about 115,000 nodes (concepts) and 340,000 links among the concepts which capture 26 different types of relations. WordNet provides opportunities for more sophisticated analysis and semantic normalization that will be subject of our future work. Synonymy information of concepts enables us to find equivalence relations between distinct triples. For example, as we expand subject node 'watcher' and predicate node 'follow' of the triple "watcher" ←"follow"→"moon" by synonyms and compare the resulting triples with the triples in the document, we find a match with "spectator"←"watch"→"moon".

Similarly we establish equivalence between the triples: "governor Patten"←"change"→"position" and "governor Patten" ←"modify"→"attitude" and triples "earthquake"←"hit"→ "Northern Iran" and "quake"←"strike"→"Northern Iran".

3.5 Construction of a Semantic Graph

We merge the logical form triples on subject and object nodes which belong to the same normalized semantic class and thus produce semantic graph, as shown in Figure 5. Subjects and objects are nodes in a graph and predicates label the relations between the nodes. Each node is also described with a set of attributes – explanatory words which are helpful for understanding the content of the node.

For each node we calculate in/out degree of a node, PageRank weight [12], Hubs and Authorities weights [8], size of weakly connected component, size of strongly connected component and many more. These statistics are used as attributes of logical form triples during the sub-graph learning process.

4. LEARNING SEMANTIC SUB-GRAPHS USING SUPPORT VECTOR MACHINES

Using linguistic procedures described in Section 3 we can generate, for each pair of documents and document summaries, the corresponding subject–predicate–object triples and associate with them a rich set of attributes, stemming from the linguistic, statistical, and graph analysis. These serve as the basis for training our summarization models.

In this section we describe the experimental set up and discuss the results.

4.1 DUC Dataset

In our experiments, we used the document collection from the Document Understanding Conference (DUC) 2002 [4], consisting of 300 newspaper articles on 30 different topics, collected from Financial Times, Wall Street Journal, Associated Press, and similar sources. Each topic contains about 10 different articles. For each article we have a 100 word human written abstract and for almost half of them also human extracted sentences, interpreted as extracted summaries. These are not used in the official DUC evaluation as they were generated by a volunteer from the research community. Nevertheless, these are useful for our analysis, as our objective is to learn characteristics of human summarization and apply the learned model to generated summaries automatically.

An average article in the data set contains about 1000 words or 50 sentences, each having 22 words. About 7.5 sentences are selected into the summary. After applying our linguistic processing, we find, on average 81 logical triples per document with 15 of them contained in extracted summary sentences.

Due to the small overlap in terms of common triples between the document and human written abstract, we had to rely on human extracted sentence summaries. In preparation for learning, we label as positive examples all subject–predicate–object triples that correspond to sentences in the summaries. Triples form other sentences are designated as negative examples.

4.2 Feature Set

Features considered for learning are logical form triples, characterized by attributes of three types. Linguistic attributes include logical form tags (subject, predicate, object), part of speech tags, depth of the linguistic node in the syntactic or logical form analysis, and about 70 semantic tags (such as gender, location name, person name, etc.). There are total 118 distinct linguistic attributes for each individual node. Fourteen additional attributes come from the semantic graph properties, some mentioned in section 3.5. Finally we include several attributes that approximate document discourse structure: the location of the sentence in the document, frequency and location of the word inside the sentence, number of different senses of the word, and related.

Each set of attributes is represented as a sparse vector of numeric values. These are concatenated into a single sparse vector and normalized to the unit length, to represent a node in the logical form triple. Similarly, for each triple the node vectors are concatenated and normalized. Thus, the resulting vectors for logical form triples contain about 466 binary and real-valued attributes. For the DUC dataset, 72 of these components have non-zero values, on average.

4.3 Learning Algorithm

This rich set of features serves as input to the Support Vector Machine (SVM) classifier [2, 7]. In the initial experiments we explored SVMs with polynomial kernel (up to degree five) and RBF kernel. However, the results were not significantly different from the SVMs with the linear kernel. Thus we continued our experiments with the linear SVMs, setting the parameter C to 2 and J to 6. Parameter C controls the tradeoff between fitting and generalization of the model, while parameter J enables the learner to weigh training errors on positive examples J times more than on negative examples.

The choice of J=6 is motivated by our assumption that the extracted summaries should aim at the higher recall, ensuring that all the human extracted sentences are included and being less sensitive to the possible noise they may be introduced.

4.4 Experimental Setup

Besides aiming at a good performance of the automatic summarizer, our important objective is to understand the relative importance of various attribute types that are available for describing the logical form triples. Thus we evaluate how adding features to the model impacts the precision and recall of extracting the logical form triples and corresponding summaries. In addition to standard precision and recall, we report for each experiment the corresponding F1 measure, defined as harmonic mean of the two statistics.

We define the learning task as a binary classification problem. We label as positive examples all subject–predicate–object triples that were extracted from the document sentences which humans selected into the summary. Triples form all other sentences are designated as negative examples. We then learn a model to discriminate between the two classes of triples.

All reported experiment statistics are micro-averaged over the instances of sentence classifications. In all of our experiments we regard the document boundaries, meaning that triples from a single document all belong either to training or test set and are never shared between the two.

Table 2: Performance of cross-topics summarization in terms of micro-average Precision, Recall and F1 measures. Results for ten-fold cross validation, for different sample size of training data.

Learning documents	Training set			Test set		
	Precision	Recall	F1	Precision	Recall	F1
10	33.48	88.44	48.58	23.05	64.67	33.99
20	31.26	86.45	45.92	24.49	68.69	36.11
50	29.42	82.53	43.37	25.75	72.81	38.04
100	28.64	79.91	42.16	26.25	73.22	38.64

Table 3: Performance of cross-topics summarization, in terms of macro-average Precision, Recall and F1 measures. Results for ten-fold cross validation, for different sample size of training data.

Attribute Set	Training set			Test set		
	Precision	Recall	F1	Precision	Recall	F1
Graph	21.63	83.40	34.35	21.35	82.40	33.91
Linguistic	26.27	75.78	39.02	25.48	73.31	37.82
Graph + Linguistic	27.28	76.66	40.24	26.97	75.96	39.81
All	28.39	78.54	41.70	27.50	76.23	40.42

Table 4: Some of the most important Subject-Predicate-Object triple attributes.

Attribute name	Attribute rank		
	1st quartile	Median	3rd quartile
Authority weight of Object node in a semantic graph	1	1	1
Size of weakly connected component of Object node in a semantic graph	2	2.5	3
Degree of Object node in a semantic graph	2	3	3
Is Object a name of a country	4	5	5
Size of weakly connected component of Subject node in a semantic graph	6	7	9
Degree of Subject node in a semantic graph	6	10.5	12
PageRank weight of Object node in a semantic graph	6	11	12
Is Object a name of a geographical location	8	13	16
Authority weight of Subject node in a semantic graph	13	18.5	23

4.5 Experiment Results

4.5.1 Impact of the Training Data Size

Our sentence classifiers are trained and tested through 10-fold cross-validation experiments, for each of several sizes of training data: 10, 20, 50, and 100 documents. Samples of documents are selected randomly and corresponding sentences used for training and testing. We always run and evaluate the resulting models on both the training and the test sets, to gain insight into the generalization aspects. From the results reported in Table 2, we observe a negative impact of small training sets on the generalization of the model.

Table 5: Performance of within-topic summarization, compared with cross-topic summarization. Comparison shows that training on topic specific yields higher F1 performance.

Learning using samples of 5 documents	Training set			Test set		
	Precision	Recall	F1	Precision	Recall	F1
Within-topic	36.49	90.63	52.03	23.60	60.05	33.89
Cross-topic	36.59	92.23	52.40	20.73	60.28	30.85

4.5.2 Impact of Different Feature Attributes

Data presented in Table 3 provides further insight into the relative importance of different attribute types, the topological graph features, the linguistic features, and the statistical and discourse attributes. Performance statistics are obtained from 10-fold cross-validation using 135 documents in the training set. Relative difference in performance has been evaluated using pair-wise t-test and it has been established that the differences between different runs are all statistically significant.

We see that using linguistic features (syntactic and semantic tags) outperforms the model relying only on the semantic graph topology. Starting with graph attributes and adding linguistic features, we experience 11.5% relative increase in the F1 measure. As new attributes are added to describe triples from additional perspectives, the performance of the classifier consistently increases. The cumulative effect of all attributes considered in the study is 19.2% relative increase in F1 measure over the baseline that uses graph attributes only.

We can also inspect the learned SVM models, i.e., the SVM normals, for the weights assigned to various attributes during the training process. We observe the relative rank of attribute weights across experiment iterations. Since the distributions of weights and corresponding attribute ranks are skewed they are best described by the median.

From Table 4 it is interesting to see that the semantic graph attributes are consistently ranked high among all the attributes used in the model. They describe the element of a triple in

Eight years after a volcano scare incited fear, anger and economic gloom in Sierra resorts, residents are nonchalant about renewed underground lava movement that is triggering thousands of tiny earthquakes.

The resort town's 4,700 permanent residents live in Long Valley, a 19-mile-long, 9-mile-wide volcanic crater known as a caldera.

The Earth's crust is being stretched apart in the region, allowing molten rock to fill half-mile-wide chambers under the caldera.

The valley was created 730,000 years ago by one of Earth's most powerful eruptions, a blast that spewed 600 times more material than the May 1980 eruption of Mount St. Helens in Washington state.

Despite the current activity, the probability of a major earthquake or a volcanic eruption in the area is "less than 1 percent each year," said David Hill, the U.S.

Geological Survey geophysicist in charge of research at Long Valley. Mono County Sheriff Martin Strelneck called such estimates "a scientific guessing game," and said area residents rarely discuss the latest swarm of earthquakes, which started in May 1989.

As a result, the Geological Survey issued a "notice of potential volcanic hazard" for Long Valley in May 1982.

That warning, coupled with jarring earthquakes, damaged tourism and aggravated a recession in the once-booming real estate market.

(a)

California's Long Valley is a 19-mile caldera created 730,000 years ago by an eruption 600 times larger than Mount St. Helens. In May 1989, new underground lava movement began triggering thousands of tiny earthquakes and raising the valley floor. Residents refuse to heed warnings, remembering a 1982 false alarm that damaged tourism and aggravated a recession. Afterward, journalists were accused of sensationalism and scientists of scaring people to get more funding. Currently, 5-10 small quakes happen daily as the Earth's crust is stretched apart and magma fills half-mile-wide chambers 4 miles under the caldera, but the probability of eruption is less than one

(b)

Figure 6: For the article on "Long Valley volcano activities" we show (a) Human extracted sentence summary and (b) 100 word human written summary.

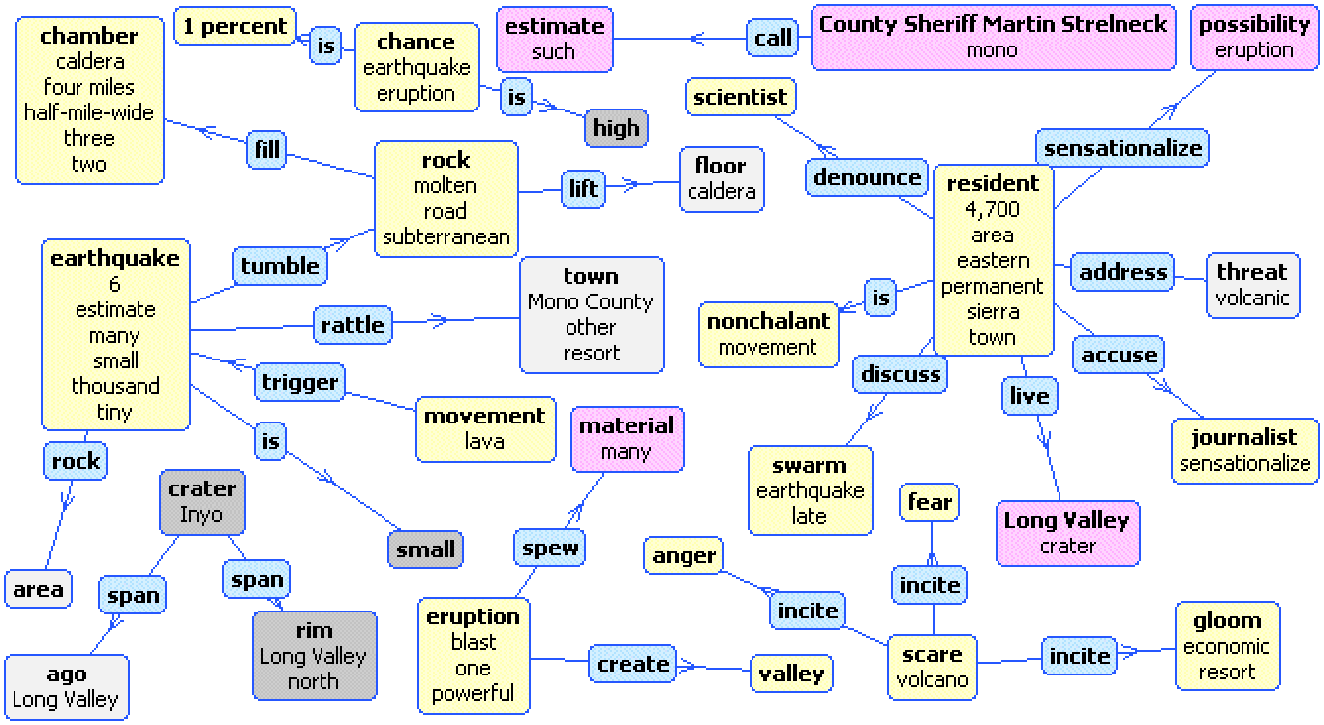

***Figure 7: Automatically generated summary (semantic graph)* from the document "Long Valley volcano activities". Subject/object nodes indicated by the light color (yellow) nodes in the graph indicate correct logical form nodes. Dark gray nodes are false positive and false negative nodes.**

relation to other entities mentioned in the text and so give overall structure of the document. For example, 'Object – authority weight' measures how other important 'hub' nodes in the graph link to it [8]. A good 'hub' points to nodes with 'authoritative' content, and a node is a good 'authority' if it is pointed to by good hubs. Subjects are hubs pointing to authorities – objects. Authority weight captures how important is the object – in how much action it is involved.

The same observation holds for experiments in which all the attributes are normalized to have a value between 0 and 1. This way we prevented the attributes with smaller values to automatically have high weights and vice versa. So all attributes had the same influence on the SVM normal. In our future work we shall study in more details SVM models with heterogeneous attribute types.

These results support our intuition that relations among concepts in the document that result from the syntactic and semantic properties of the text are important for summarization. Interestingly, feature attributes that most strongly characterize non-summary triples are mainly linguistic attributes describing gender, position of the verb, as being inside the quotes, position of the sentence in the document, word frequency, and similar – the latter few attributes are typically used in statistical approaches to summary extraction.

4.5.3 Topic Specific Summarization

For each of 30 topics there are 5 documents on the average with extracted sentence summaries. We used these documents to learn topic specific summaries. We performed 'leave one out' cross validation over all 30 topics. More precisely, for each topic, we take all the documents for training, except one. We learn the classifier on the selected documents (all triples extracted from the documents) and test the classifier on one that is left out. This is repeated for each document in the test and the performance is averaged over all the cross validation runs.

Table 5 shows the performance that is higher from the performance of the topic independent summaries, when the same sample of training data is used. While the data size for topic specific summarization is small and thus does not allow generalization, the results may be indicative of the ability to use the method on a rather small dataset and capture the topic specific trends.

4.6 Sentence Extraction

Extracted logical form triples are used to identify the appropriate sentences for inclusion into the summary. We apply a simple decision rule by which a sentence is included in the summary if at least one of the logical triples from the sentence is nominated by the learner as the summary triple. As logical form triples can be shared by multiple summary sentences, as well as sentences that may not be appropriate for summaries, this additional step introduces further noise. Applying F1 measure to the extracted sentences we find that the micro-average F1 value is about 2% lower relative to those for the triples themselves.

In Figure 5 we show a sentence selection summary of the document about the Long Valley volcano activities. The original document is about 1000 words in length. Human generated sentence selection summary contains 200 words (Figure 6, (a)). Next, we show a 100 word human written summary of the same document (Figure 6, (b)). Comparing the text of human extracted and human generated summaries we can appreciate the complexity of automatic summary generation and evaluation of the experimental systems.

Figure 5 and Figure 7 show semantic graphs for the document and automatically generated summary using our method. Light shaded nodes represent summary nodes – nodes that were generated by the triples extracted from the sentences humans selected into the summary (Figure 7). Dark shaded nodes were generated by the triples from non-summary sentences.

5. RELATED WORK

Over the past decades, research in text summarization has produced a great volume of literature and methods [13, 15]. However, even the simplified definition of document summaries as document extracts, as proposed by Luhn [10], does not simplify the task of creating comprehensible and useful document surrogates. One can group the efforts into those based on heuristics [5, 10, 14], others based on machine learning techniques [9, 16] and those combining the two. One common theme of research is the depth of the linguistic analysis undertaken in the summarization effort. While most of this work stays at the shallow parsing level, our approach is unique in three main respects. It introduces an intermediate, more generic layer, of text representation within which the structure and content of the document and summary are captured. This direction of research has been outline by Spark Jones in [15] as still unexplored avenue. Second we apply machine learning technique using features that capture semantic structure, i.e., concepts and relations, in contrast to previous attempts in which linguistic features are of finer granularity, i.e., keywords and noun phrases [9, 16]. Finally, the intermediate semantic graph representation opens up new areas of explorations in which the captured semantic structure itself can serve as a document surrogate and provide means for document navigation.

6. CONCLUSIONS

We presented a novel approach of document summarization by generating semantic representation of the document and applying machine learning to extract a sub-graph that corresponds to the semantic structure of a human extracted document summary. Experiments on the DUC 2002 data show that adding attribute types to the logical form features help increase the performance of the learnt model, as evidenced by the increase in the micro-average F1 measure. Compared to human extracted summaries we achieve on average recall of 75% and precision of 30%. Our future work will involve explorations of alternative semantic structures on additional data sets, including human generated abstracts.

7. REFERENCES

[1] Bowman, B., Debray, S. K., and Peterson, L. L. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst., 15,* 5 (Nov. 1993), 795-825, 1993.

[2] Burges, C.J.C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2 (2): 121–167, 1998.

[3] Corston-Oliver, S.H. and Dolan, B. Less is more: eliminating index terms from subordinate clauses. *Proceedings of the 37th Conference on Association for Computational Linguistics*, College Park, Maryland, 1999.

[4] Document Understanding Conference (DUC), 2002. http://tides.nist.gov/.

[5] Edmunton, H.P. New Methods in Automatic Extracting. *Journal of the ACM* (JACM), 16(2):264-285, 1969.

[6] Fellbaum, Ch. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[7] Joachims T. Making large-scale support vector machine learning practical. In: Bernard Sch□olkopf, Christopher J. C. Burges, Alexander J. Smola, eds., *Advances in kernel methods: Support vector learning*. The MIT Press, 1999.

[8] Kleinberg, J.M. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5): 604–632, 1999.

[9] Kupiec, J., Pederson, J. & Chen, F. A Trainable Document Summarizer. *Proceedings of SIGIR'95*, 1995.

[10] Luhn, H.P. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159-165, 1959.

[11] Mitkov, R. Robust pronoun resolution with limited knowledge. *Proceedings of the 18th International Conference on Computational Linguistics* COLING'98/ACL'98, 869-875, Montreal, 1998.

[12] Page, L., Brin, S., Motwani, R. and Winograd T.. *The* PageRank citation ranking: Bringing order to the web. *Digital libraries project report*, Stanford University, 1998.

[13] Paice, C.D. Constructing literature abstracts by computer: Techniques and prospects. *Information processing and Management*, 26:171-186, 1990.

[14] Salton, G., Alan, J. and Buckley, C. Approaches to passage retrieval in full text information systems. In *Proceedings of SIGIR'93*, 49-58, 1993.

[15] Sparck-Jones, K(1997) Summarizing: Where are we now? Where should we go? In I. Mani and M. Maybury (eds.), Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain.

[16] Teufel, S. & Moens, M. Sentence extraction as a classification task. In I. Mani and M. Maybury (eds.), Proceedings of the *ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, 1997.