Extracting Summary Sentences Based on the Document Semantic Graph

Jure Leskovec

Natasa Milic-Frayling

Marko Grobelnik

January 31, 2005

MSR-TR-2005-07

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Extracting Summary Sentences Based on the Document Semantic Graph

Jure Leskovec
Carnegie Mellon University, USA
Jozef Stefan Institute, Slovenia
Jure.Leskovce@ijs.si

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

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

ABSTRACT

We present a method for extracting sentences from an individual document to serve as a document summary or a pre-cursor to creating a generic document abstract. We apply syntactic analysis of the text that produces a logical form analysis for each sentence. We use subject–object–predicate (SOP) triples from individual sentences to create a semantic graph of the original document and the corresponding human extracted summary. Using the Support Vector Machines learning algorithm, we train a classifier to identify SOP triples from the document semantic graph that belong to the summary. The classifier is then used for automatic extraction of summaries from test documents. Our experiments with the DUC 2002 and CAST datasets show that including semantic properties and topological graph properties of logical triples yields statistically significant improvement of the micro-average F1 measure for both the extraction of SOP triples that correspond to the semantic structure of extracts and the extraction of summary sentences. Evaluation based on ROUGE shows similar results for the extracted summary sentences.

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. To automate the process of abstracting, researchers generally rely on a two phase process. First, key textual elements, e.g., keywords, clauses, sentences, or paragraphs are extracted from text using linguistic and statistical analyses. In the second step, the extracted text may be used as a summary. Such summaries are referred to as 'extracts'. Alternatively, textual elements can be used to generate new text, similar to the human authored abstract.

Automatic generation of texts that resemble human abstracts presents a number of challenges. While abstracts may include portions of document text, it has been shown that authors of abstracts often rewrite the text, interpreting the content and fusing the concepts. In the study by Jing [6] of 300 human-written summaries of news articles, 19% of summary sentences did not have matching sentences in the document. The remainder of summary sentences overlapped with a single sentence content in 42% of cases. This included matches through paraphrasing and syntactic transformation, implying that the number of perfectly aligned matches would be even lower. Other studies show that the number of aligned sentences varies significantly from corpus to corpus. For the set of 202 computational linguistic papers used by Teufel and Moens [18] the perfect alignment is observed for only 31.7% of abstract sentences. That figure rises to 79% in 188 technical papers in [9]. Thus, if the automatic summarization methods are to take advantage of the texts from the document it is important to investigate alignment on the sub-sentence level, e.g., at the level of clauses as investigated by Marcu [12]. Comparing the meaning of clauses in the document and corresponding abstracts, by employing human subjects, Marcu [12] showed that in order to create an abstract from extracted text one may need to start with a pool of extracted clauses with a total length 2.76 times larger than the length of the resulting abstract.

This implies that relevant concepts, carrying the meaning, are scattered across clauses. Starting with a hypothesis that the main functional elements of sentences and clauses are Subjects, Objects, and Predicates, we ask whether identifying and exploiting links among them could facilitate the extraction of relevant text. Thus, we devise a method that creates a semantic graph of a document, based on logical form triples subject–predicate–object (SPO), and learns a relevant sub-graph that could be used for creating summaries.

In order to establish the plausibility of this approach we first focus on learning to automate human extracts. We assess how well the model can extract the substructure of the graph that corresponds to the extracted sentences. This substructure is then the basis for extracting the relevant text from the document. Restricting the evaluation to sentence extraction we gain a good understanding of the effectiveness of the approach and learnt model. Essentially we decouple the evaluation of the learning model from the issues of text generation that arises in the creation of abstracts.

In this paper we present results from our experiments on two data sets, CAST [4] and a part of DUC 2002 [3], equipped with human extracted summaries. We demonstrate that the feature attributes related to the connectivity of the semantic graph and linguistic properties of the graph nodes significantly contribute to the performance of our summary extraction model. With this understanding we set solid foundations for exploring similar learning models for document abstraction.

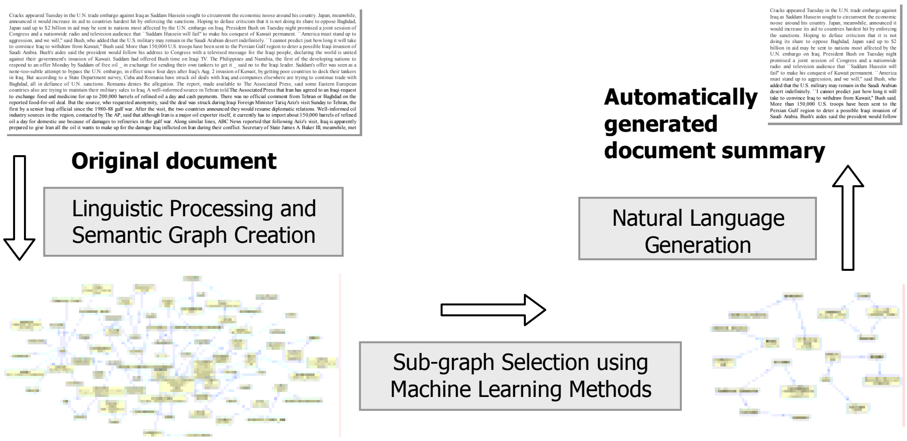

Original document
Linguistic Processing and Semantic Graph Creation
Sub-graph Selection using Machine Learning Methods
Natural Language Generation
Automatically generated document summary

Figure 1. Summarization procedure based on semantic structure analysis.

In the following sections we describe the procedure that we use to generate the semantic graphs and define feature attributes for the learning model. We present the results of the experiments and discuss how they can guide the future work.

2. SEMANTIC GRAPH GENERATION

In this study we create a novel representation of the document content that relies on the 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 linguistic properties of the nodes in the triples to build semantic graphs for both documents and corresponding summaries.

We expect that the graph of the extracted summary would capture essential semantic relations among concepts and that the resulting structure could be found within the corresponding document semantic graph. Thus, we reduce the problem of summarization to acquiring machine learning models for mapping between the document graph and the graph of a summary.

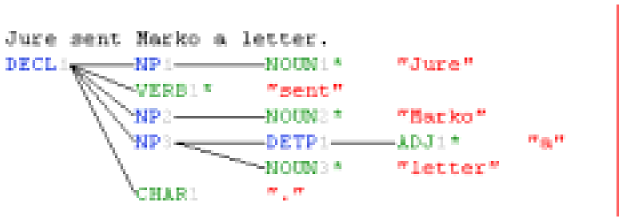
We generate a semantic graph in three steps:

- Syntactic analysis of the text – We apply deep syntactic analysis to document sentences, using NLPWin linguistic tool [2][5], and extract logical form triples.
- Co-reference resolution – We identify co-references for named entities through the surface form matching and text layout analysis. Thus we consolidate expressions that refer to the same named entity.
- We merge the resulting logical form triples into a semantic graph and analyze the graph properties. The nodes in our graphs correspond to Subjects and Objects. A link between them corresponds to a Predicate.

In our research we investigated semantic graphs that involved pronominal reference resolution and semantic normalization. However, initial experiments showed that using anaphora resolution which achieved 80% accuracy and WordNet [20] for synonym normalization yields marginal improvement in the performance of the summary extractor. Thus, for the sake of clarity and simplicity we present the method using minimal post-processing of the NLPWin output through co-reference resolution.

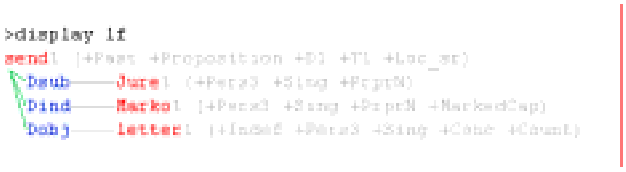
2.1 Linguistic Analysis

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.



```
Jure sent Marko a letter.
DECL1  NP1    NOUN1*   "Jure"
       VERB1* "sent"
       NP2    NOUN2*   "Marko"
       NP3    DETP1    ADJ1*   "a"
              NOUN3*   "letter"
       CHAR1  "."
```

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



```
>display lf
send1 (+Past +Proposition +D1 +T1 +Loc_sr)
 Dsub    Jure1 (+Pers3 +Sing +PrprN)
 Dind    Marko1 (+Pers3 +Sing +PrprN +MarkedCap)
 Dobj    letter1 (+Indef +Pers3 +Sing +Conc +Count)
```

Figure 3. Logical form for the sentence

The logical form 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 (e.g., "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.

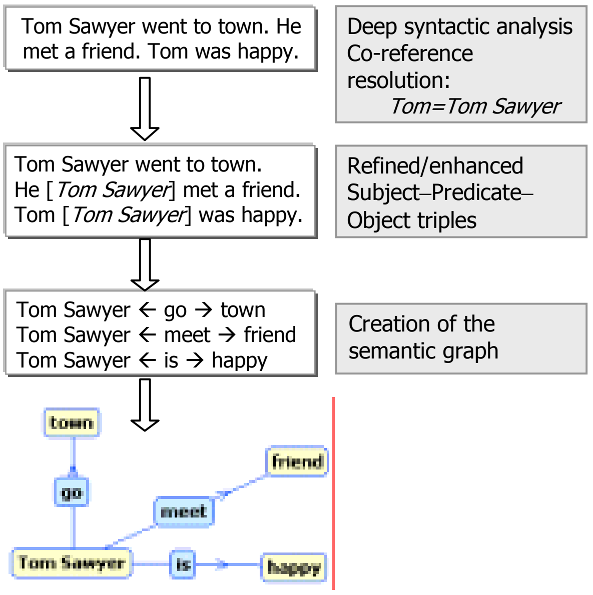

Tom Sawyer went to town. He met a friend. Tom was happy.
Deep syntactic analysis
Co-reference resolution:
Tom=Tom Sawyer
Tom Sawyer went to town.
He [Tom Sawyer] met a friend.
Tom [Tom Sawyer] was happy.
Refined/enhanced Subject–Predicate–Object triples
Tom Sawyer ← go → town
Tom Sawyer ← meet → friend
Tom Sawyer ← is → happy
Creation of the semantic graph
town
go
friend
meet
Tom Sawyer
is
happy

Figure 4. Process of creating a semantic graph.

Figure 4 outlines the main processes. Identified logical form triples are linked into a graph based on common nodes. Figure 5 shows an example of a semantic graph for an entire document.

2.2 Co-reference Resolution for Named Entities

It is common that terms with different surface forms refer to the same entity in the same document. 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 which reduces the number of terms that need to be examined for co-reference resolution. 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", "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. This approach is similar to the ones explored in related research [14] and has proven to be effective in the context of our study, yielding better learning models.

2.3 Construction of the Semantic Graph

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

For each node in a semantic graph we calculate the number of topological properties. These are later used as attributes of logical form triples during the sub-graph learning process. The full set of features used in the learning process is given in section 3.2

3. LEARNING SEMANTIC SUB-GRAPHS USING SUPPORT VECTOR MACHINES

Using linguistic procedures described in Section 2 we can generate, for each pair of document and document summary, the corresponding set of subject–predicate–object triples and associate them with a rich set of attributes, coming from linguistic, statistical, and graph analysis. These serve as the basis for training our summarization models.

3.1 Data Sets

We run our experiments on two data sets: a subset of the DUC2002 dataset and CAST collection.

3.1.1 DUC2002 Data set

We use the DUC2002 document collection from the Document Understanding Conference (DUC) 2002 [3]. For our experiments we use training part of DUC 2002, which consists of 300 newspaper articles on 30 different topics, collected from Financial Times, Wall Street Journal, Associated Press, and similar sources. Almost half of these documents have human extracted sentences, interpreted as extracted summaries. These are not used in the official DUC evaluation since DUC is primarily focused on generating abstracts. Thus, we cannot make a direct comparison with DUC systems performance. However, the data is useful for our objective of exploring various aspects of our approach.

On average, an article in the DUC data set contains about 1100 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. In preparation for learning, we label as positive examples all subject–predicate–object triples that correspond to sentences in the human extracted summaries. Triples form other sentences are designated as negative examples.

3.1.2 CAST Data set

CAST corpus [4] contains texts from the Reuters Corpus annotated with information that can be used to train and evaluate automatic summarization methods. Four annotators marked 15% of document sentences as *essential* and additional 15% as *important* for the summary. However the distribution of documents across assessors has been rather arbitrary and for some documents we have up to three sets of sentence selections while for others only one. For that reason we decided to run our experiments on the set of 89 documents annotated by a single assessor, Annotator 1. We run experiments that model separately extraction of short (15%) summaries, represented by sentences marked as essential, and longer (30%) summaries, which include both sentences marked as essential and sentences marked as important.

An average length article in the CAST data set contains about 528 words or 29 sentences, each having 18 words. The assessor selected on average about 6 sentences for short summaries and additional 6 for longer summaries. After applying our linguistic

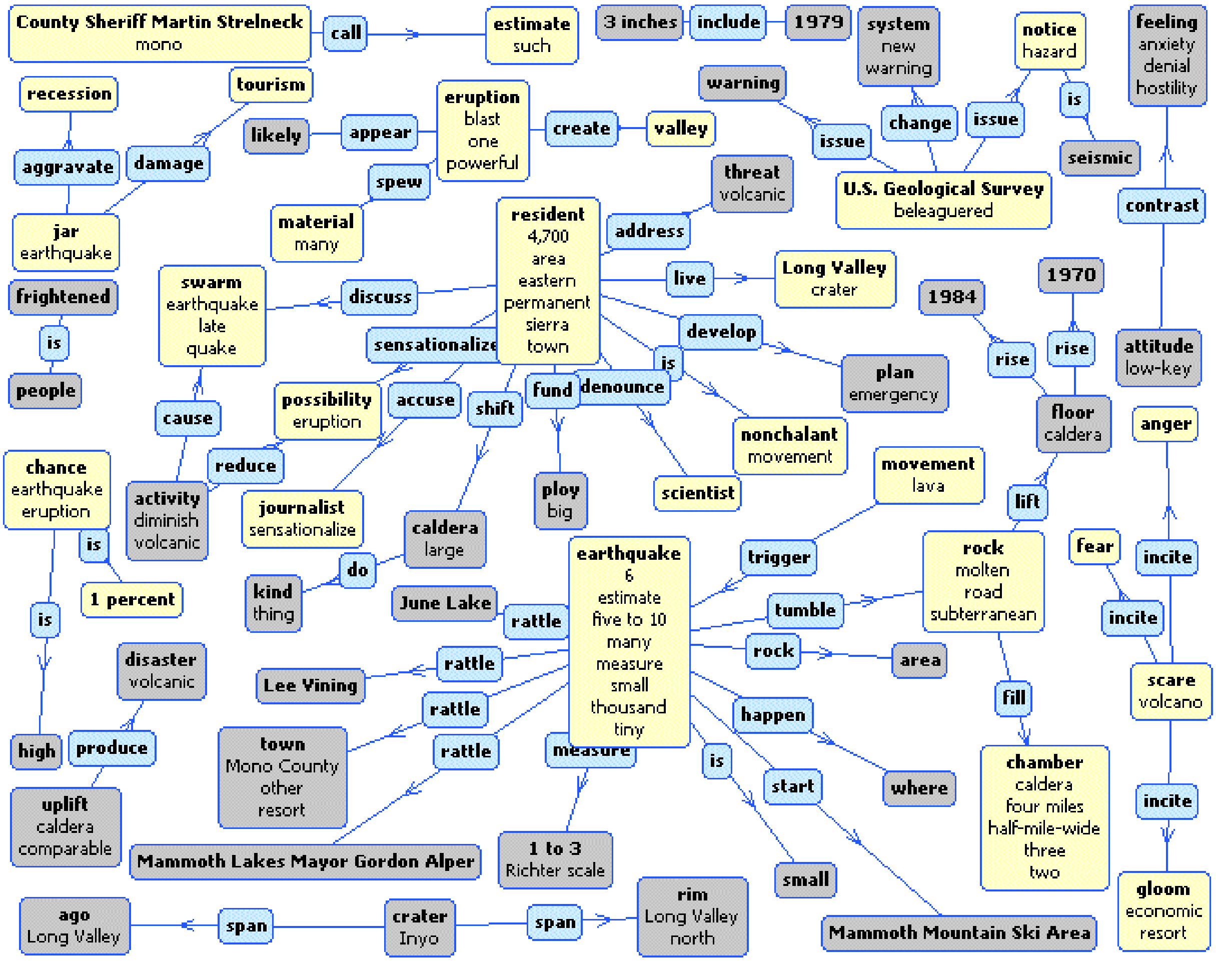


Figure 5. Full semantic graph of the DUC 2002 document "Long Valley volcano activities". Subject/object nodes indicated by the light color (yellow) nodes in the graph indicate summary nodes. Gray nodes indicate non-summary nodes. We learn a model for distinguishing between the light and dark nodes in the graph.

processing, we find on average 41 logical form triples per document with 6 or 12 of them included in extracted sentences for short and longer summaries, respectively.

3.2 Feature Set

As features for the learning process, we consider logical form triples characterized by three types of attributes:

- *Linguistic attributes* which include logical form tags (subject, predicate, object), part of speech tags, and about 70 semantic tags (such as gender, location name, person name, etc.). There are total 118 distinct linguistic attributes for each node.
- *Semantic graph* attributes describing properties of the graph. For each node we calculate the number of incoming and outgoing links, Hubs and Authorities [8] and PageRank [15] weights. We also include the statistics on the number nodes reachable by 2, 3 and 4 hops away respectively, and the total of reachable nodes. We consider both the directed and undirected versions of the semantic graph when calculating these statistics. There are total 14 attributes calculated from the semantic graph.
- *Document discourse structure* is approximated by several attributes: the location of the sentence in the document and the triple in the sentence, 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 binary and real-valued numbers. 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. The resulting vectors for logical form triples contain about 372 binary and real-valued attributes. For the DUC dataset, 69 of these components have non-zero values, on average. For the CAST dataset we find 327 attributes total with 68 non-zero values per triple on average.

3.3 Learning Algorithm

This rich set of features serves as input to the Support Vector Machine (SVM) classifier [1][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

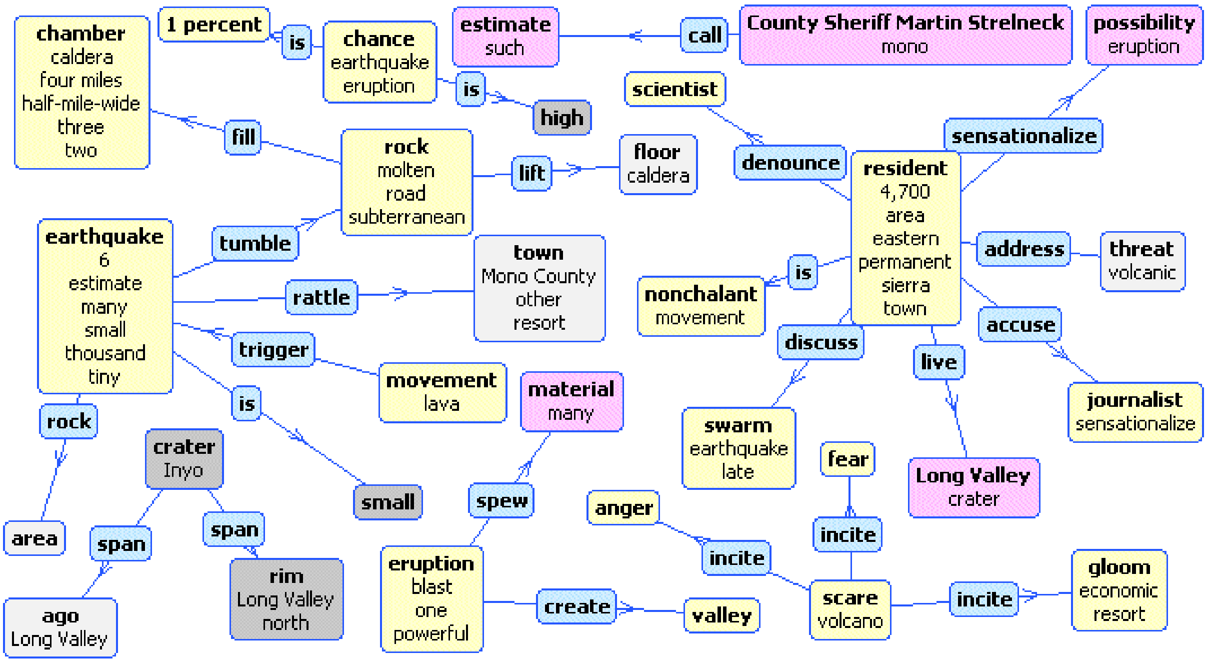


Figure 6. Automatically generated summary (semantic graph) from the document "Long Valley volcano activities". Subject/object nodes indicated by the light color (yellow) nodes in the graph indicate correct logical form nodes. Dark gray nodes are false positive and false negative nodes.

from the SVMs with the linear kernel. Thus we continued our experiments with the linear SVMs.

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 from all other sentences are designated as negative examples. We then learn a model that discriminates between these two classes of triples.

3.4 Experimental Setup

We evaluate performance of both, the extraction of semantic structure elements, i.e., logical form triples, and the extraction of document sentences. We use extracted logical form triples 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 it contains at least one triple identified by the learning algorithm. We accumulate the summaries to satisfy the length criteria. All reported experiment statistics are micro-averaged over the instances of logical triple and sentence classifications, respectively.

One important objective of our research is to understand the relative importance of various attribute types that describe the logical form triples. Thus we evaluate how adding features to the model impacts the precision and recall of extracted logical form triples and corresponding summaries. We report the standard precision and recall and their harmonic mean – the F1 score. All the experiments are run using stratified 10-fold cross-validation, where samples of documents are selected randomly and corresponding sentences (triples) are used for training and testing. We take into account the document boundaries and therefore the triples from a single document all belong either to the training or test set and are never shared between the two.

We always run and evaluate the resulting models on both the training and the test sets, to gain an insight into the generalization of the model. When evaluating summaries, we are also interested in the coverage of the human extracts achieved by our extracted summaries. In instances where we miss to extract the correct sentence, we still wish to assess whether the automatically extracted sentence is close in content to the ones that we missed. For that we calculate the overlap between the automatically extracted summaries and human extracted summaries using ROUGE [10], the measure adopted by DUC as the standard for assessing the summary coverage. ROUGE is a recall oriented measure, based on n-gram statistics that has been found highly correlated with human evaluations. We use ROUGE n-gram(1,1) statistics and restrict the length of the automatically generated summary to be the same as of the human sentence extract.

4. EXPERIMENT RESULTS

Tables 1–3 summarize the results of the sentence extraction based on the learned SVM classifier for the DUC and CAST datasets. Precision, recall and F1 measures for the extraction of triples are very close to the performance of extracted sentences and therefore we do not present them separately.

4.1.1 Impact of Different Feature Attributes

Performance statistics presented in Tables 1 to 3 provides insight into the relative importance of different attribute types, the graph topological properties, the linguistic features, and the statistical and discourse attributes.

Table 1: Performance of sentence extraction on the DUC2002 extracts, in terms of macro-average Precision, Recall and F1 measures and Rouge score. Results for stratified ten-fold cross validation.

Attribute set	Training set			Test set			
	Precision	Recall	F1	Precision	Recall	F1	Rouge
Sentence position and terms	65.08	92.14	76.28	28.77	37.27	32.48	0.69
Triple and sentence position	31.29	53.38	39.45	31.12	53.34	39.32	0.71
Graph attributes	28.26	62.99	39.02	27.58	61.67	38.11	0.73
Linguistic attributes	25.79	62.48	36.51	20.79	51.87	29.69	0.78
Position + Linguistic	30.74	67.33	42.21	28.66	63.23	39.44	0.76
Position + Graph	34.44	65.37	45.11	33.67	64.39	**44.22**	**0.83**
Position + Graph + Linguistic	34.25	71.40	46.29	31.85	66.77	43.13	0.82

Table 2: Performance of the sentence selection on the CAST 15% extracts (essential sentences), in terms of macro-average Precision, Recall and F1 measures and Rouge score. Results for the stratified ten-fold cross validation.

Attribute set	Training set			Test set			
	Precision	Recall	F1	Precision	Recall	F1	Rouge
Sentence position and terms	85.54	87.43	86.42	30.32	25.14	27.49	0.59
Triple and sentence position	33.07	65.69	44.99	32.54	64.54	43.27	0.62
Graph attributes	20.92	59.52	30.95	19.82	56.85	29.39	0.66
Linguistic attributes	35.95	57.10	44.12	21.34	32.83	25.87	0.62
Position + Linguistic	39.89	74.59	51.89	34.31	63.41	44.53	0.73
Position + Graph	33.70	72.63	46.04	32.47	70.92	**44.54**	0.73
Position + Graph + Linguistic	40.43	77.40	53.12	33.83	64.35	44.34	**0.74**

Table 3: Performance on the CAST 30% extracts (essential and important sentences), in terms of macro-average Precision, Recall and F1 measures and Rouge score. Results for the stratified ten-fold cross validation.

Attribute set	Training set			Test set			
	Precision	Recall	F1	Precision	Recall	F1	Rouge
Sentence position and terms	87.97	84.27	86.08	43.24	33.68	37.86	0.59
Triple and sentence position	44.62	59.44	50.97	43.67	58.42	49.98	0.68
Graph attributes	38.42	67.42	48.95	36.80	65.85	47.22	0.67
Linguistic attributes	45.96	80.41	58.41	40.22	70.84	51.31	**0.73**
Position + Linguistic	50.57	74.18	60.14	43.92	64.25	52.18	0.72
Position + Graph	45.10	70.60	55.04	43.47	67.26	52.81	0.71
Position + Graph + Linguistic	51.04	75.00	60.74	44.45	65.57	**52.98**	0.72

The first row of each table shows the baseline model where we use only sentence position and sentence terms for learning the model. In all cases we observe very good performance of the baseline on training set, but the model does not generalize well – has poor performance on the test set. The Rouge score of baseline is also quite low. For comparison we also generated another set of baseline summaries by taking first sentences in each document. Over all datasets Rouge score of these summaries was additional 0.10 lower than of the baseline obtained using machine learning.

For the all datasets, the performance statistics are obtained from the 10-fold cross-validation. Relative difference in performance has been evaluated using pair-wise t-test and it has been established that the differences between different runs are statistically significant.

From Table 1 we see that including semantic graph attributes consistently improves recall and thus the F1 score. Starting with only linguistic attributes and adding information about position, we experience 9.75% absolute increase in the F1 measure. As new attributes are added to describe the triple from additional perspectives, the performance of the classifier consistently increases. The cumulative effect of all attributes considered in the study is 26.5% relative increase in F1 measure over the baseline that uses only sentence terms and position attributes. The model which uses information about position of the triple and the structure of semantic graph performs best both in F1 and Rouge scores.

Table 4: Some of the most important Subject–Predicate–Object triple attributes for DUC experiments

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

In terms of Rouge measure, linguistic features (syntactic and semantic tags) outperform the model which relies only on the semantic graph. For linguistic attributes we also observe a discrepancy between F1 and Rouge score. Linguistic attributes score low on F1 but usually relatively high on Rouge. On the other hand, for position attributes we observe the reverse effect – good F1 and low Rouge score.

We make similar observations on CAST dataset (tables 2 and 3). We see that using position and graph attributes gives very good performance in terms of F1 and Rouge measures. We observe that using only semantic graph attributes does not give a very good performance. While the size of sentence extracts in DUC and CAST are similar, DUC documents are much longer, contain more logical triples, and therefore have semantic graphs that are better connected. We manually inspected CAST semantic graphs and observed that they are not so well connected and appear less helpful for summarization.

4.1.2 Observations from the SVM Normal.

We also inspect the learned SVM models, i.e., the SVM normal, for the weights assigned to various attributes during the training process. We normalize each attribute to have a value between 0 and 1. This way we prevent the attributes with smaller values to automatically have higher weights. We then observe the relative rank of attribute weights over 10 folds. 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 the attributes used in the model. They describe the elements of a triple in relation to other entities mentioned in the text and capture the overall structure of the document. For example, 'Authority weight of Object node' measures how other important 'hub' nodes in the graph link to it. A good 'hub' points to nodes with 'authoritative' content, and a node has a high 'authority' if it is pointed to by good hubs. In our graph representations, subjects are hubs pointing to authorities – objects and thus the authority weight captures how important is the object, i.e., in how much actions, described by predicates, it is involved.

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.

5. RELATED WORK

Over the past decades, research in text summarization has produced a great volume of literature and methods. For overview and insights into the state-of-the-art we refer to [16][17] and comment on the work that relates to several aspects of our approach. While most of the past work stays in the realm of shallow text parsing level and statistical processing, our approach is unique in that it combines two aspects: (1) it introduces an intermediate layer of text representation within which the structure and the content of both the document and summary are captured and (2) it uses machine learning to identify elements of the semantic structures, i.e., concepts and relations, as oppose to learning from linguistic features of finer granularity, such as keywords and noun phrases [9][18] or yet, complete sentences [13]. We also note that the semantic graph representation opens possibilities for novel types of document surrogates, focused not on reading but navigation through the document on the basis of captured concepts and relations.

Graph based methods. Application of graph representation in summarization has been applied by Mihalcea [13] by treating individual sentences as nodes in the graph and establishing links among the sentences based the content overlap. In addition to the difference in the text granularity level at which the graph is applied, the method in [13] does not involve learning. It selects sentences by setting the threshold on the scores associated with the graph nodes.

Most similar to our approach to constructing the semantic graph is the method by Vanderwende et al. [19] aimed at generating event-centric summaries. The method uses the same linguistic tool, NLPWin to obtain logical form triples from sentences but constructs the semantic graph in a rather different way. In order to capture text about events Vanderwende et al. [19] treat Predicates as nodes in the graph, together with Subjects and Objects while the links between the nodes are inherited from the logical form analysis. More precisely, the atomic structure of the graph is a triple (Node$_i$, relation/link, Node$_j$), where relation is a syntactic tag such as: direct object, location, time, and similar. For example, the graph would contain ("Marko", Subject, "Send"), ("Send", Object, "Letter"), ("Send", Time, "Wednesday"). In our representation the elementary structure is (“Marko”, “Send”, “Letter”). Therefore, the statistical properties of the graph and link weight propagation have different meaning and effect. Similarly to Mihalcea [13], Vanderwendte et al. [19] do not apply learning to select substructures but set the score threshold for selection of logical form triples.

Both methods [13] and [19] are applied in the context of generating abstracts and their encouraging results lead us to believe that further evaluation of our method will show similar results.

In their work Mani & Bloedorn [11] and Kupiec et al. [9] applied several learning algorithms to the set of features that were in the previous research applied in an adhoc manner to select text for summarization (sentence location, statistical measures of term prominence, similarity between sentences, presence of proper names or certain syntactic features in the sentence, etc.). The significant contribution of our work is in widening the type of features for learning to those that capture both the structure and the content and enhance our understanding of the role that these structural elements play in modeling sentence extraction for summarization.

6. SUMMARY AND FUTURE WORK

We presented a novel approach to document summarization which generates a semantic representation of the document and applies machine learning to extract semantic sub-structure suitable for creating summaries. We evaluated our approach on a simpler problem of sentence extraction for document summaries. This enabled us to focus on the characteristics of the learning model and investigate the relative importance of feature attributes used in learning. Experiments on the two data sets show that the attributes which capture properties of the document semantic structure play an important role in the sentence selection process.

Our approach, has a number of advantages over methods used so far. Semantic structure based on the logical form enables us to extract triples that correspond to sub-clauses of document sentences. This provides a good foundation for collecting text segments that would be useful for abstract creation and multi-document summarization.

Furthermore, the rich set of linguistic and graph attributes enable the learning algorithm to select the set of attributes that best model the summarization process for a particular set of documents and a particular performance measure. For example, we noticed that for training data with shorter summaries linguistic features play more significant role in optimizing the performance than the structure features. That is reversed in the situation where we have longer summaries and longer documents, for which the semantic structure is richer and more informative.

Our future work will involve explorations of alternative semantic structures on additional data sets and a wider set of summarization problems, including human generated abstracts and cross document summaries.

7. REFERENCES

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

[2] Corston-Oliver, S.H. and Dolan, B. Less is more: eliminating index terms from subordinate clauses. *ACL*, 1999.

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

[4] Hasler, L., Orasan, C. and Mitkov, R. Building better corpora for summarization. *Corpus Linguistics 2003*.

[5] Heidron, E.G. Intelligent Writing Assistance. In *Handbook of Natural Language Processing*, Eds. Dale, R., Moisl, H. and Somers, G. Marcel Dekker, 2000.

[6] Jing, H. Using Hidden Markov Modeling to Decompose Human-Written Summaries. *Computational Linguistics* 4, 28, 527-543, 2002.

[7] Joachims, T. Making large-scale support vector machine learning practical. *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. In *Proceedings of SIGIR'95*, 1995.

[10] Lin, J.C. and Hovy, E. H. Automatic evaluation of summaries using n-gram co-occurrence statistics. *Human Language Technology Conference*, Edmonton, 2003.

[11] Mani, I. and Bloedorn, E. Machine Learning of Generic and User-Focused Summarization. *AAAI* 1998.

[12] Marcu, D. The automatic construction of large-scale corpora for summarization research. *SIGIR* 1999.

[13] Mihalcea, R. Graph-based ranking algorithms for sentence extraction, applied to text summarization. *ACL* 2004.

[14] Nenkova, A. and McKeown, K. References to Named Entities: a Corpus Study. *HLT-NAACL* 2003.

[15] 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.

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

[17] Sparck-Jones, K. Summarizing: Where are we now? Where should we go? *ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization,* Madrid, Spain, 1997.

[18] Teufel, S. and Moens, M. Sentence extraction as a classification task. *ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization,* Madrid, Spain, 1997.

[19] Vanderwende, L., Banko, M., and Menezes, A. Event-Centric Summary Generation. *DUC* 2004.

[20] Fellbaum, C. WordNet: An Electronic Lexical Database. MIT Press, 1998.