Impact of Linguistic Analysis on the Semantic Graph Coverage and Learning of Document Extracts

Jure Leskovec[13], Natasa Milic-Frayling[2], Marko Grobelnik[3]

[1]Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, Pennsylvania 15213, USA
[2]Microsoft Research Ltd
Roger Needham Bldg., 7 J J Thomson Avenue, Cambridge CB3 0FB, United Kingdom
[3]Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
jure@cs.cmu.edu, natasamf@microsoft.com, marko.grobelnik@ijs.si

Abstract

Automatic document summarization is a problem of creating a document surrogate that adequately represents the full document content. We aim at a summarization system that can replicate the quality of summaries created by humans. In this paper we investigate the machine learning method for extracting full sentences from documents based on the document semantic graph structure. In particular, we explore how the Support Vector Machines (SVM) learning method is affected by the quality of linguistic analyses and the corresponding semantic graph representations. We apply two types of linguistic analysis: (1) a simple part-of-speech tagging of noun phrases and verbs and (2) full logical form analysis which identifies Subject-Predicate-Object triples, and then build the semantic graphs. We train the SVM classifier to identify summary nodes and use these nodes to extract sentences. Experiments with the DUC 2002 and CAST datasets show that the SVM based extraction of sentences does not differ significantly for the simple and the sophisticated syntactic analysis. In both cases the graph attributes used in learning are essential for the classifier performance and the quality of extracted summaries.

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. The ultimate objective of summarization systems is to enable automatic abstracting of the document text, with all the properties that humans bring to that process. However, that task stretches beyond text analysis to domain knowledge, inference, and language generation. Most of the research has therefore been concerned with methods for text processing and extraction of textual segments that approximate human abstracts. Recently, document summarization research has been given a significant boost by the Document Understanding Conference (DUC 2002), which provides an experimentation framework and a forum for exchanging ideas.

Recent work by (Vanderwende et al. 2004) and (Leskovec et al. 2005) demonstrates the use of rich document semantic structure for document summarization. Both represent the document text as a semantic graph that consists of nodes representing terms and edges capturing the relations among terms. They use the graph properties to identify nodes that are useful for creating document extracts and abstracts. This approach is a significant departure from the traditional way of qualifying summary sentences using features such as the location of a sentence in a document or the appearance of specific key words in sentences, and applying heuristic scoring or machine learning techniques.

(Leskovec et al. 2005) use a machine learning technique to identify sub-structures of the document semantic graph that are found in human created sentence extracts. These are then used to create document summaries by extracting full sentences from the original document text. (Vanderwende et al. 2004), on the other hand, apply a task specific scoring technique, aimed at capturing event summaries, and use a score threshold to identify sub-structures for generating the summary text.

In both instances, the semantic graphs are based on a sophisticated linguistic analysis which identifies subject–predicate–object of individual sentences. This raises a question that we address in this paper: what role does linguistic analysis play in learning graph sub-structures and optimizing the quality of extracted sentences? Is it possible to relax the complexity of sentence analysis and still obtain decent summary extracts?

Our experiments show that the use of a less sophisticated linguistic analysis and the corresponding semantic representations does not affect the performance of sentence extraction. In fact, in some instances it leads to summaries with a better coverage of manually created summaries. The resulting semantic graphs typically cover a larger percentage of the document and summary text, without diluting the semantic graph properties. Indeed, the graph attributes used in learning remain essential for the performance of the classifier and sentence selection.

Our results thus open the door to more flexible and economical ways of building semantic graphs for the purpose of summarization. The proven robustness of the SVM learning algorithm seems as a good basis for tackling more challenging problems, such as cross-document summarization and generation of document abstracts.

In the following sections we introduce the basic concepts of the model, referring to the related work as appropriate. We discuss the experiment set up and the results of the sub-graph learning experiments. We conclude by summarizing our results and outlining the future work.

Background Research

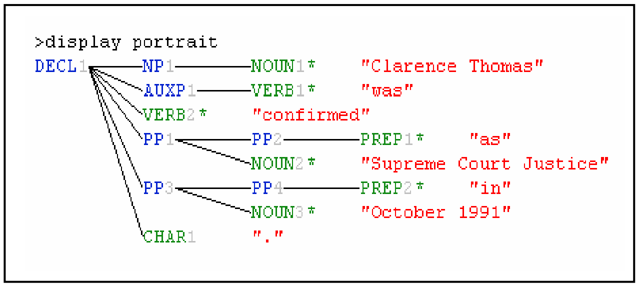
Most of the past research in automated document summarization stays within shallow text parsing and statistical processing. The latter typically involves scoring and selecting candidate sentences using heuristics based on sentence location, statistical measures of term prominence, similarity between sentences, presence of proper names or certain syntactic features in the sentence, etc. (Mani 1998) and (Kupiec 1995) took a more systematic approach and applied machine learning to the set of similar attributes.

Alternative approaches have been taken, for example, by (Salton, et al. 1994) who applied information retrieval techniques to identify the topical structure of documents from the similarities between paragraphs and used it to create summaries. Recently (Mihalcea 2004) combined sentence similarity with graph representation and obtained very encouraging results towards approximating human abstracts. In that model graph nodes correspond to individual sentences and the links capture sentence similarities. Each node is assigned a score based on similarity statistics and graph properties, which then facilitates the selection of nodes and summary sentences.

While these approaches focus on sentence or paragraph features, (Marcu 1999) compares the meaning of clauses in documents and human created abstracts using human subjects. He showed that, in order to compose an abstract from clauses extracted from original documents, one may have to start with a pool of clauses almost three times larger than the length of the resulting abstract. This implies that concepts that exist in both abstracts and original documents are scattered across clauses. Therefore, in order to take a step towards document abstracting, it is important to work with sub-sentence textual units. For this reason, the work by (Vanderwende et al. 2004) and (Leskovec et al. 2005) is of particular interest. In the following section we discuss NLPWin (Heidron 2000), the linguistic tool they both use to perform linguistic analysis and generate semantic graph representations.

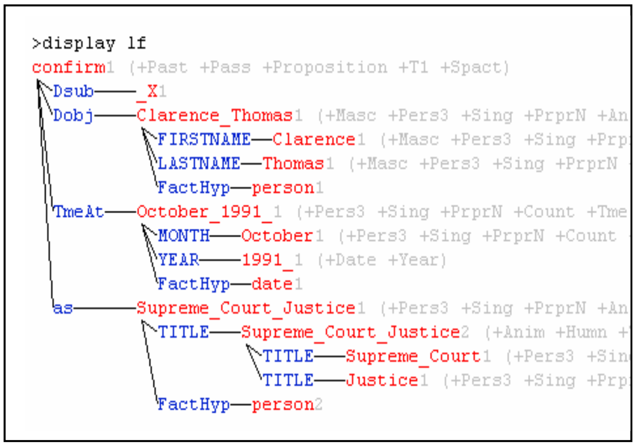
Linguistic Analysis

NLPWin is a natural language processing tool which provides deep syntactic and partial semantic analysis of text. NLPWin segments the text into individual sentences and processes them in several steps, starting with the lexical and morphological analysis of individual linguistic tokens and creating a parse tree. We are particularly interested in the next phase, referred to as Portrait, in which the parse tree is analyzed to identify the correct placement of the modifier phrases (Figure 1). In the final phase, NLPWin produces the main functional elements of the sentence, Subject–Predicate–Object, referred to as logical form triples (Figure 2).



```
>display portrait
DECL1────NP1─────────NOUN1*    "Clarence Thomas"
         AUXP1───────VERB1*    "was"
         VERB2*    "confirmed"
         PP1─────────PP2─────────PREP1*    "as"
                     NOUN2*    "Supreme Court Justice"
         PP3─────────PP4─────────PREP2*    "in"
                     NOUN3*    "October 1991"
         CHAR1     "."
```

Figure 1. Parse tree from the NLPWin Portrait analysis for the sentence: "*Clarence Thomas was confirmed as Supreme Court Justice in October 1991*".



```
>display lf
confirm1 (+Past +Pass +Proposition +T1 +Spact)
  Dsub────_X1
  Dobj────Clarence_Thomas1 (+Masc +Pers3 +Sing +PrprN +An
            FIRSTNAME──Clarence1 (+Masc +Pers3 +Sing +Prp
            LASTNAME──Thomas1 (+Masc +Pers3 +Sing +PrprN
            FactHyp──person1
  TmeAt───October_1991_1 (+Pers3 +Sing +PrprN +Count +Tme
            MONTH───October1 (+Pers3 +Sing +PrprN +Count
            YEAR────1991_1 (+Date +Year)
            FactHyp──date1
  as──────Supreme_Court_Justice1 (+Pers3 +Sing +PrprN +An
            TITLE───Supreme_Court_Justice2 (+Anim +Humn +
                      TITLE───Supreme_Court1 (+Pers3 +Sin
                      TITLE───Justice1 (+Pers3 +Sing +Prp
            FactHyp──person2
```

Figure 2. NLPWin Logical Form shows the Subject, Object, Predicate ("confirm") nodes. In the parenthesis are semantic tags of the nodes.

We use the Portrait analysis to extract simple linguistic structures, such as noun phrases (designated as NP) and verbs. In one of the semantic graph representations we include modifiers of nouns, e.g., adjectives, as graph nodes while in others we use only normalized head nouns, e.g., for "Supreme court" we retain the head noun "court".

For the Logical Form Analysis we use the Heads of Subject, Object, and Predicate to create 'atomic' graph structures that consist of node triples or head noun nodes and Predicate edges.

Semantic Graph Construction and Analysis

Similarly to (Leskovec et al. 2005), we create document semantic graphs in the following three steps:

- Syntactic analysis of text – We use two levels of syntactic analysis from the NLPWin Portrait output: (1)

noun-phrase and verb detection, with part-of-speech information and (2) the logical form triple.

- Co-reference resolution – In text, different surface forms may refer to the same entity. We identify co-references for named entities, such as names of people, places, and companies by post-processing the NLPWin output.
- Semantic graph fusion – We merge the graphs of individual sentences based on the match of their nodes in normalized form and analyze the graph properties.

After the creation of the semantic graphs we determine node attributes that are used by the SVM classifier. In the final stage, when the relevant nodes are identified, we apply a sentence extraction algorithm to obtain the final selection of summary sentences.

Creation of the Semantic Graph

From the NLPWin analysis we use three different sets of linguistic features:

(**ANV**) – Adjectives, nouns, and verbs
(**NPV**) – Head nouns (from noun phrases) and verbs
(**LF**) – Heads of logical form triples.

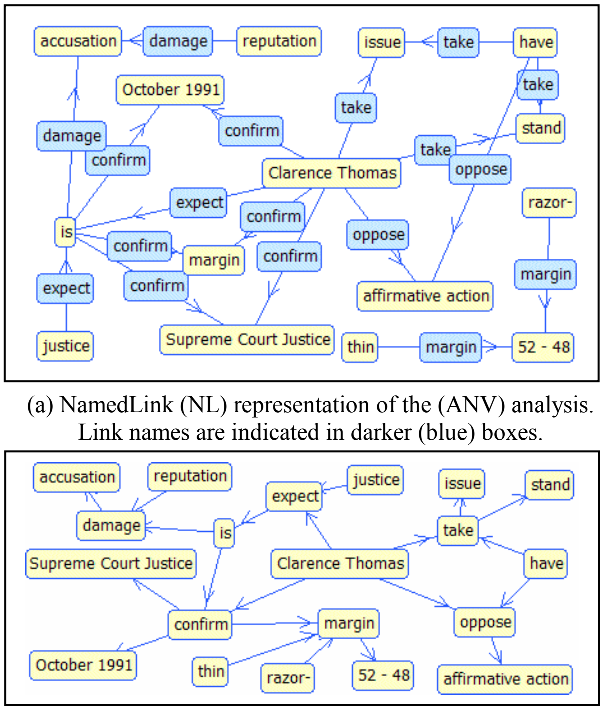
As elementary graph structures, we consider triples of linguistic terms, modeled after the logical form triples. The triples can be incorporated into the graphs in two ways. Firstly, in case of LF triples, the Predicate, i.e., verb node, can be viewed as a link between the Subject and Object nodes. Alternatively, a verb node can be treated as a node itself. In the latter case, the links are not named but could inherit linguistic relations ("Dsubj", "Dobj", in Figure 2) as in the representation by (Vanderwende et al. 2004). We thus have two graph representations for each of the three sets of linguistic features above:

(**NL**) – **NamedLink** representation, where links correspond to linguistic features. In (NPV) the link refers to a verb. In the case of (ANV), where we explicitly used modifiers as nodes, links can also be nouns.

(**N**) – **Nodes** representation, where links have no labels. All the linguistic terms are represented as nodes of the graph.

Figures 3 and 4 show the NamedLink and Nodes representations for (ANV) and (LF), for the paragraph: *"Clarence Thomas was confirmed as Supreme Court Justice in October 1991 by a razor-thin margin of 52-48. Thomas, who has opposed affirmative action, has not taken public stands on other key issues. His reputation was damaged by accusations of sexual harassment. As the youngest justice he is expected to be on the court for decades."* [From DUC2002 Data set]

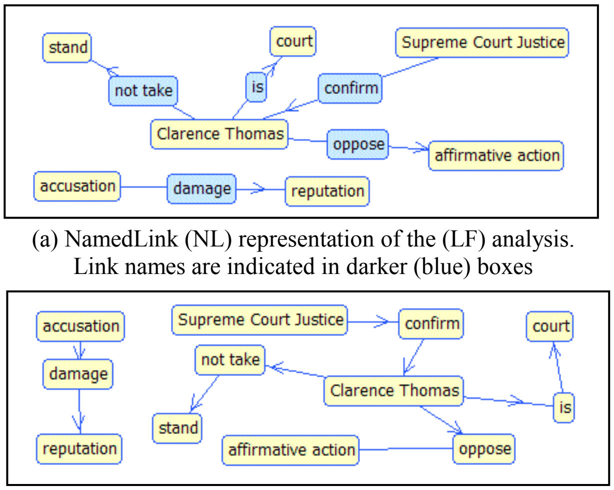
Finally, sentences may not always produce full LF. However, such sentences may be plausible candidates for a summary sentence. For that reason we consider including not only triples but also pairs of linguistic terms. Inclusion of pairs, however, is not suitable for the NamedLink semantic graph representation because of the incomplete paths (links without nodes). Thus we consider pairs only within the Nodes representation.



(a) NamedLink (NL) representation of the (ANV) analysis. Link names are indicated in darker (blue) boxes.

(b) Nodes (N) representation of the (ANV) analysis.

Figure 3. Two graph representations, NamedLink and Nodes, for the basic linguistic analysis: attributes, nouns, and verbs.



(a) NamedLink (NL) representation of the (LF) analysis. Link names are indicated in darker (blue) boxes

(b) Nodes (N) representation of the (LF) analysis.

Figure 4. NamedLink and Nodes representation for the Logical Form analysis, showing heads of the Subject, Predicate and Object terms.

Characterizing Graph Nodes

For each node in the graph representation, we specify a number of attributes that characterize the node. These attributes are used by the learning algorithm to differentiate the nodes that are useful for extracting document summaries from those that are not. Similarly to (Leskovec at al. 2005) we use three types of attributes: linguistic attributes, graph attributes, and text location statistics, which approximate a discourse structure of the document.

Linguistic attributes. For each node, NLPWin provides part-of-speech tags and about 70 semantic tags (e.g., gender, location name, person name, etc.). We use in total 118 distinct linguistic attributes for each node.

Graph attributes. For each node in a semantic graph we calculate the number of incoming and outgoing links, Hubs and Authorities and PageRank weights (Page et al. 1998). We include statistics on the number of nodes reachable in 2, 3, and 4 hops away, respectively, and the total number of reachable nodes. Altogether there are 14 distinct graph attributes calculated for each node in the semantic graph.

Document structure attributes. For each elementary sub-structure, a triple or a pair, we consider the location of the corresponding sentence in the document and the location of the triple or the pair within the sentence. These values are assigned to the corresponding nodes. We also determine the frequency and location of the word inside the sentence, and the number of different senses of the word.

Learning from the Graphs

We train the linear Support Vector Machine (SVM) classifier to identify sub-structures of the semantic graphs, i.e., the node triples and pairs, that are useful for identifying sentences for document extracts. The SVM learner is applied to a rich set of attributes described above. In fact, each type of attributes, i.e., linguistic attributes, graph attributes, and document structure attributes, are represented as sparse vectors of binary and real-valued features. Depending on the experiment, we concatenate selected attribute type vectors into a single feature vector and normalize it to the unit length. The concatenated vectors represent individual nodes in the graph. Similarly we obtain feature vectors for sub-structure elements, i.e., triples and pairs, by concatenating and normalizing the individual node feature vectors.

All our experiments are conducted with the same SVM settings: the parameter C is set to 1 and J to 4. Here C controls the tradeoff between the fit to the data and the generalization of the model. The parameter J, on the other hand, enables us to weigh training errors on positive examples J times more than on negative examples. We set J to 4, aiming at a higher recall so that human extracted sentences are more likely to be included, possibly at the expense of lower precision.

Extracting Sentences

Once the SVM classifies the sub-structure units, i.e., triples and pairs, into summary and non-summary ones, we use them to extract sentences from the text. For a newly processed document, we score each sentence based on the confidence scores of summary triples (and pairs) it contains. In particular, we *add up confidence scores of constituent summary triples and pairs,* ignoring the scores of non-summary ones. From sentences that score above the classification threshold we select a predefined number of sentences for the summary. In our experiments, that number is determined as the *average number of sentences* in the document extracts of a particular data set.

Experiments

In order to evaluate the impact of various linguistic analyses on the graph properties and classification performance, we conducted experiments with several datasets and used standard evaluation measures.

Data Sets

DUC2002 Data Set. DUC 2002 is one of the document collections provided by the Document Understanding Conference (DUC). Since we are looking at the problem of extracting summary sentences, we use a training part of the DUC 2002 data, 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. 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 the objectives of our research.

Cast Data Set. CAST corpus (Hasler et al. 2003) 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 an additional 15% as *important* for the summary. However, the distribution of documents across assessors was rather arbitrary, which lead to an uneven number of user judgments across documents. For that reason, in our experiments we use the set of 89 documents annotated by a single assessor, Annotator 1. We run separate experiments for extraction of short summaries – dataset CAST-15%, that include sentences marked as essential, and longer summaries – dataset CAST-30%, containing sentences that are marked as essential or important. An average length article in the CAST data set contains about 29 sentences. The assessor selected on average 6 sentences for short summaries and additional 6 for longer summaries.

Graph Characteristics

Text Coverage. It is important to analyze how many sentences are, in fact, covered by semantic graphs since only those sentences that are represented by the graph are considered for learning. Furthermore, only those test sentences that produce linguistic nodes that are recognized by the system can be classified. Thus, the linguistic procedures themselves define the upper limit on the

Semantic Graph Structure	Covered summary sentences [%]	Covered non-summary sentences [%]
ANV with Triples	93.4	86.6
ANV with Triples + Pairs	**98.6**	94.6
NPV with Triples	73.4	63.6
NPV with Triples + Pairs	**94.0**	83.3
LF with Triples	80.0	69.8
LF with Triples + Pairs	**90.6**	87.4

Table 1. Coverage of the summary sentences on DUC dataset. A sentence is covered if it contains at least one triple (triple+pair).

Data Set	Linguistic Analysis	Graph Representation	Sub-structure units
DUC CAST-15% CAST-30%	ANV NPV LF	NamedLinks Nodes	Triples Triples+Pairs (for Nodes)

Table 2. Experiment design involves four different dimensions. A mixture of node triples and node pairs was used to create only the Nodes representation of the semantic graph.

performance of the summary extraction systems. Table 1 shows, as expected, that the coverage of sentences increases with a simpler linguistic analysis and inclusion of node pairs.

Experiment Design and Evaluation Measures

Our aim is to investigate the impact of different levels of linguistic analysis on the performance of the classifier. To that end, we performed an exhaustive set of experiments that involves three data sets and, for each of them, three models of linguistic analysis (ANV), (NPV) and (LF), using NamedLink and Nodes representations of semantic graphs (see Table 2). We also added pairs of nodes to the Nodes representation to see the impact of smaller substructures on the sentence extraction performance.

For each experiment, we ran 10-fold cross-validation and performed a t-test to determine statistical significance of observed differences in performance. Furthermore, during the learning procedure we varied the types of attributes associated with the graph nodes (see Table 5, for example).

Evaluation Measures. For each experiment we present the standard precision and recall measures that capture the percentage of correctly extracted summary sentences for the fixed length summary. We also report the corresponding F1 measure, defined as a harmonic mean of the two statistics. All the statistics are micro-averaged over the instances of sentence classification.

DUC 2002				
Semantic Structure	Prec.	Recall	F1	Rouge
ANV	0.39	0.40	**0.40**	**0.67**
NPV	0.37	0.39	0.38	0.65
LF	0.40	0.40	**0.40**	0.64

CAST-30%				
Semantic Structure	Prec.	Recall	F1	Rouge
ANV	0.43	0.57	0.49	**0.67**
NPV	0.41	0.66	0.50	0.63
LF	0.42	0.67	**0.52**	**0.66**

CAST-15%				
Semantic Structure	Prec.	Recall	F1	Rouge
ANV	0.44	0.44	0.44	**0.64**
NPV	0.48	0.47	**0.48**	0.61
LF	0.45	0.44	0.45	0.61

Table 3. Comparison of the three linguistic analyses, using only node triples, for the summaries of predefined length. Graph representation is NamedLinks. We used all attributes for learning.

In addition, we determine the word coverage of human extracts achieved by our extracted summaries. Even if the system fails to extract the correct sentence, it is important to assess whether the extracted sentence is close in content to the correct one. We thus calculate the overlap between automatically extracted summaries and human extracted summaries using ROUGE, a measure adopted by DUC. ROUGE is recall oriented, based on n-gram statistics, and found to be highly correlated with human evaluations.

Experiment Findings

Robustness of the SVM Classifier. SVM classifier learns equally well from both the simple and more sophisticated linguistic analysis. Table 3 shows typical results on the three data sets that we used. The sentence extraction does not suffer from simpler linguistic analysis. In fact, on DUC 2002 data, learning from the simplest linguistic model (ANV - adjective, noun, verb) performs equally as well as the LF representation, according to the F1 measure. The ROUGE score for ANV, in fact, shows a statistically significant increase over the LF score.

Robustness of the Graph Properties. Increasing the coverage of summary sentences, by relaxing the linguistic analysis and combining pairs and triples of linguistic nodes, does not seem to dilute the value of graph properties or negatively affect sentence extraction. Table 4 shows the results for DUC 2002 data. The F1 measure, which captures the sentence level performance, is not significantly affected. The ROUGE score, measuring the word level

DUC 2002				
Semantic Structure	**Prec.**	**Recall**	**F1**	**Rouge**
ANV with Triples	0.31	0.62	**0.41**	**0.67**
ANV with Triples+Pairs	0.30	0.63	**0.41**	0.66
NPV with Triples	0.29	0.62	0.40	0.65
NPV with Triples+Pairs	0.30	0.64	**0.41**	**0.67**
LF with Triples	0.39	0.39	**0.39**	0.64
LF with Triples+Pairs	0.38	0.39	0.38	**0.65**

Table 4. Comparison of the three linguistic analyses, with and without node pairs included in the graph. The results are for the Nodes only representation, using all attributes for learning.

agreement with human extracts, is slightly improved by adding node pairs.

Interaction of the Linguistic and Graph Attributes. Table 5 shows a subset of experiments that verify the individual contribution of attribute types to the performance of the summarizer. Adding graph attributes to document structure (referred to as Position) and linguistic attributes systematically helps sentence extractions across different linguistic analyses. For example, the difference in the ROUGE scores for LF and ANV (Table 5) and the two sets of attributes is statistically significant. Similar behavior is observed with the Cast datasets. This confirms that semantic graphs associated with simpler analyses (e.g., ANV) are amenable to learning of relevant semantic nodes and can produce improved sentence extracts (Table 5).

Concluding Remarks

In this paper we investigated several aspects of the document summarization technique that involves creating a semantic graph of the document and training a SVM classifier to identify relevant structures for summary extracts. We demonstrated that it is possible to achieve comparable and often improved summary extracts by applying a simple linguistic analysis and building semantic graphs based on resulting triples and pairs.

Reducing the complexity of syntactic analysis yields a wider coverage of document and summary texts. On one hand we expect increase in performance since more examples are used in SVM training and more of test summary sentences are processed by the system. On the other, semantic graphs with simple and possibly less distinctive nodes may cause degradation of the learning process and poorer summaries. Thus, it was important to establish empirically which of these factors prevail. We show that the SVM is robust and learns well from simpler linguistic analyses while graph attributes remain essential in improving the node classification and summary extraction across all of the experiments. We expect that robustness of the SVM will prove useful in addressing more challenging tasks of cross document summarization and creation of document abstracts.

DUC 2002				
Semantic Struct. Attributes	**Prec.**	**Recall**	**F1**	**Rouge**
ANV with Pos + Ling	0.27	0.58	0.37	0.65
ANV with Pos + Ling + Graph	0.39	0.40	**0.40**	**0.67**
NPV with Pos + Ling	0.37	0.35	0.36	0.62
NPV with Pos + Ling + Graph	0.37	0.39	**0.38**	**0.65**
LF with Pos + Ling	0.26	0.61	0.36	0.62
LF with Pos + Ling + Graph	0.40	0.40	**0.40**	**0.64**

Table 5. Comparison of the three linguistic analyses, with varied attribute sets for graph nodes. The results are shown for the NamedLink graph (from triples), for fixed length summaries.

References

Document Understanding Conference. http://tides.nist.gov

Hasler, L., Orasan, C. and Mitkov, R. 2003. Building better corpora for summarization. *Corpus Linguistics'03*.

Heidron, E.G. 2000. Intelligent Writing Assistance. *In Handbook of Natural Language Processing*. Eds. Robert Dale, Herman Moisl, and Harold Somers, Marcel Dekker.

Kupiec, J., Pederson, J. & Chen, F. 1995. A *Trainable Document Summarizer. In Proc. of SIGIR'95.*

Leskovec, J., Milic-Frayling, N., and Grobelnik, M. 2005. Extracting Summary Sentences Based on the Document Semantic Graph. Microsoft Technical Report TR-2005-07.

Mani, I. and Bloedorn, E. 1998. Machine Learning of Generic and User-Focused Summarization. *AAAI'98*.

Marcu, D. 1999. The automatic construction of large-scale corpora for summarization research. *SIGIR'99*.

Mihalcea, R. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization.*ACL'04*.

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

Salton, G., Allan J., Buckley C., & Singhal A. 1994. Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts. *Science*, 264.

Vanderwende, L., Banko, M., and Menezes, A. 2004. Event-Centric Summary Generation. *DUC'04*.