# On the Effectiveness of Using Sentence Compression Models for Query-Focused Multi-Document Summarization

*Yllias Chali* *Sadid A. Hasan*
University of Lethbridge, Lethbridge, AB, Canada
chali@cs.uleth.ca, hasan@cs.uleth.ca

ABSTRACT
This paper applies sentence compression models for the task of query-focused multi-document summarization in order to investigate if sentence compression improves the overall summarization performance. Both compression and summarization are considered as global optimization problems and solved using integer linear programming (ILP). Three different models are built depending on the order in which compression and summarization are performed: 1) *ComFirst* (where compression is performed first), 2) *SumFirst* (where important sentence extraction is performed first), and 3) *Combined* (where compression and extraction are performed jointly via optimizing a combined objective function). Sentence compression models include lexical, syntactic and semantic constraints while summarization models include relevance, redundancy and length constraints. A comprehensive set of query-related and importance-oriented measures are used to define the relevance constraint whereas four alternative redundancy constraints are employed based on different sentence similarity measures using a) cosine similarity, b) syntactic similarity, c) semantic similarity, and d) extended string subsequence kernel (ESSK). Empirical evaluation on the DUC benchmark datasets demonstrates that the overall summary quality can be improved significantly using global optimization with semantically motivated models.

KEYWORDS: Sentence compression, query-focused multi-document summarization, integer linear programming (ILP).

*Proceedings of COLING 2012: Technical Papers*, pages 457–474,
COLING 2012, Mumbai, December 2012.

## 1 Introduction and Related Work

Text summarization is a good way to compress large amount of information into a concise form by selecting the most important information and discarding redundant information (Mani and Maybury, 1999). Query-focused multi-document summarization aims to create a summary from the available source documents that can answer the requested information need (Chali and Hasan, 2012). Extraction-based automatic summarization has been a common practice over the years for its simplicity (Edmundson, 1969; Kupiec et al., 1995; Carbonell and Goldstein, 1998; Lin, 2003; Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011). Extraction of the most important sentences to form a summary can degrade the summary quality if there exists a longer sentence with partly relevant information to prevent inclusion of other important sentences (due to summary length constraint) (Martins and Smith, 2009). Sentence compression can be a good remedy for this problem where the task can be viewed as a single-sentence summarization (Jing, 2000; Clarke and Lapata, 2008). Sentence compression[1] aims to retain the most important information of a sentence in the shortest form whilst being grammatical at the same time (Knight and Marcu, 2000, 2002; Lin, 2003). Previous researches have shown that sentence compression can be used effectively in automatic summarization systems to produce more informative summaries by reducing the redundancy in the summary sentences (Jing, 2000; Knight and Marcu, 2002; Lin, 2003; Daumé III and Marcu, 2005; Zajic et al., 2007; Madnani et al., 2007; Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011). However, most of these researches either focused on the task of single document summarization and generic summarization or did not consider global properties of the sentence compression problem (Clarke and Lapata, 2008). Due to the vast increase in both the amount of online data and the demand for access to different types of information in recent years, attention has shifted from single document and generic summarization[2] toward query-based multi-document summarization. On the other hand, sentence compression can achieve superior performance if it can be treated as an optimization problem and solved using integer linear programming (ILP) to infer globally optimal compressions (Gillick and Favre, 2009; Clarke and Lapata, 2008). ILP has recently attracted much attention in the natural language processing (NLP) community (Roth and Yih, 2004; Clarke and Lapata, 2008; Punyakanok et al., 2004; Riedel and Clarke, 2006; Denis and Baldridge, 2007). Gillick and Favre (2009) proposed to extend their ILP formulation for a concept-based model of summarization by incorporating additional constraints for sentence compression. However, to the best of our knowledge, there has not been a single research that deeply investigates the potential of using ILP-based sentence compression models for the task of query-focused multi-document summarization. In this paper, we accomplish this task by considering both compression and summarization as global optimization problems.

The sentence compression models used in the existing automatic summarization systems mostly exploit various lexical and syntactic properties of the sentences (Knight and Marcu, 2002; Mcdonald, 2006; Clarke and Lapata, 2008; Cohn and Lapata, 2008; Galanis and Androutsopoulos, 2010). A recent work has shown that discourse segmentation could be incorporated in a sentence compression system which can aid automatic summarization (Molina et al., 2011).

[1] Although most of the works on sentence compression are mainly related to the English language, researchers have also worked on sentence compression related to languages other than English (Molina et al., 2011; Filippova, 2010; Bouayad-Agha et al., 2006). Our work is applied to the English language. However, we believe that the proposed techniques can be applicable to other languages provided that the lexical, syntactical and semantic properties of the corresponding language are considered.

[2] A generic summary includes information which is central to the source documents whereas a query-oriented summary should formulate an answer to the user query (Goldstein et al., 1999).

Lin (2003) showed that pure syntactic-based compression does not improve a generic summarization system. A most recent work has shown that sentence compression can achieve better performance if semantic role information can be incorporated into the model (Yoshikawa et al., 2012). Inspired by their work, we recast their formulation as an ILP for sentence compression with semantic role constraints. We build three different ILP-based sentence compression models: 1) a bigram language model with lexical and syntactic constraints (derived from Clarke and Lapata (2008)), 2) the bigram language model with a topic signature modeling function (Lin and Hovy, 2000), and 3) the bigram language model with semantic role constraints (Yoshikawa et al., 2012). We choose to build them since the variation of these models were shown to achieve better results comparable to the state-of-the-art techniques (Clarke and Lapata, 2008; Yoshikawa et al., 2012). We perform a rigorous study to analyze the effectiveness of using these sentence compression models to generate query-focused summaries. For this study, we compose three different models depending on the order to perform sentence compression and extraction: 1) *ComFirst*, 2) *SumFirst*, and 3) *Combined*. The main motivation behind building these models is that we intend to study if the order of performing compression and extraction can affect the overall performance of the query-focused multi-document summarization. Martins and Smith (2009) argued that the two-step "pipeline" approaches such as *ComFirst* and *SumFirst* might often fail to select global optimal summaries.

Query-focused extractive multi-document summarization generally needs three essential criteria to be satisfied (McDonald, 2007): 1) Relevance: to contain informative sentences relevant to the given query, 2) Redundancy: to not contain multiple similar sentences, and 3) Length: should follow a fixed length constraint. We define a global optimization model that uses ILP to infer optimal summaries. The existing ILP formulations to the summarization task mostly rely on relevance and redundancy functions (such as word-level cosine similarity measure, word bigrams) that are primitive in nature (McDonald, 2007; Gillick and Favre, 2009; Martins and Smith, 2009). The major limitation of these approaches is that they do not consider the sequence of words (i.e. word ordering). They ignore the syntactic and semantic structure of the sentences and thus, cannot distinguish between "The police shot the gunman" and "The gunman shot the police". The researchers speculate that the better the relevance and redundancy functions could be, the more the solutions would be efficient (Gillick and Favre, 2009). In the proposed optimization framework, we incorporate a comprehensive set of query-related and importance-oriented measures to define the relevance function. We employ four alternative redundancy constraints based on different sentence similarity measures using a) cosine similarity, b) syntactic similarity, c) semantic similarity, and d) extended string subsequence kernel (ESSK). We propose the use of syntactic tree kernel (Moschitti and Basili, 2006), shallow semantic tree kernel (Moschitti et al., 2007), and a variation of the extended string subsequence kernel (ESSK) (Hirao et al., 2003) to accomplish the task. Our empirical evaluation on the DUC benchmark datasets demonstrate the effectiveness of applying sentence compression for the task of query-focused multi-document summarization. The results also show that the quality of the generated summaries vary based on the use of alternative redundancy constraints in the optimization framework.

## 2 ILP-based Sentence Compression Models

An ILP is a constrained optimization problem, where both the cost function and constraints are linear in a set of integer variables (McDonald, 2007; Clarke and Lapata, 2008). In this section we describe three ILP-based sentence compression models which we apply for the task of query-focused multi-document summarization. Our first model is a bigram language model

derived from the work of Knight and Marcu (2002); Clarke and Lapata (2008). Our second model is close in spirit rather different in content to Clarke and Lapata (2008). In this model, we combine the bigram language model with a corpus-based topic signature modeling approach of Lin and Hovy (2000). Our first two models include various lexical and syntactical constraints based on the work of Clarke and Lapata (2008). In the third model, we add a set of semantically motivated constraints into the bigram language model based on the work of Yoshikawa et al. (2012).

## 2.1 Bigram Language Model

According to Clarke and Lapata (2008), the sentence compression problem can be formally defined as follows. Let $S = w_1, w_2, \cdots, w_n$ is an original sentence in a document. To represent the words to be included in the compressed version of this sentence, we define a set of indicator variables $\delta_i$ that are set to 1 if $i$-th word is selected into the compression, and 0 otherwise. To make decisions based on word sequences (rather than individual words), we define additional indicator variables $a_i$ (that are set to 1 if $i$-th word starts the compression, and 0 otherwise), $b_i$ (that are set to 1 if $i$-th word ends the compression, and 0 otherwise), and $c_{ij}$ (that are set to 1 if sequence $w_i, w_j$ is present in the compression, and 0 otherwise). Now the inference task is solved by maximizing the following objective function (that includes the overall sum of the decision variables multiplied by their log-transformed corpus bigram probabilities) (Clarke and Lapata, 2008):

$$Maximize \sum_i a_i \cdot P(w_i|start) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} \cdot P(w_j|w_i) + \sum_i b_i \cdot P(end|w_i) \quad (1)$$

such that $\forall i, j \in \{1 \cdots n\}$ :

$$\delta_i, a_i, b_i, c_{ij} \in \{0, 1\} \quad (2)$$

$$\sum_i a_i = 1 \quad (3)$$

$$\delta_j - a_j - \sum_{i=1}^{j} c_{ij} = 0 \quad (4)$$

$$\delta_i - \sum_{j=i+1}^{n} c_{ij} - b_i = 0 \quad (5)$$

$$\sum_i b_i = 1 \quad (6)$$

$$\sum_i \delta_i \geq l \quad (7)$$

$$\sum_{i:w_i \in verbs} \delta_i \geq 1 \quad (8)$$

$$\delta_i = 1 \quad (9)$$
$$\forall i : w_i \in personal\ pronouns$$

$$\delta_i = 0 \quad (10)$$
$$\forall i : w_i \in words\ in\ parentheses$$

$$\delta_i - \delta_j = 0 \quad (11)$$
$$\forall i, j : w_j \in possessive\ mods\ of\ w_i$$

The objective function in Equation 1 is maximized to find the optimal target compression where "start" and "end" denote $w_0$ and $w_n$, respectively. The above ILP formulation incorporates

various constraints. The first constraint states that the variables are binary. The later constraints are defined to disallow invalid bigram sequences in the compression. Constraint 3 states that exactly one word can start a compression. Constraint 4 and Constraint 5 are responsible to ensure correct bigram sequences, whereas Constraint 6 denotes that exactly one word can end the compression. On the other hand, Constraint 7 forces the compression to have at least $l$ words. We add some additional constraints (Constraint 8 to Constraint 11) from Clarke and Lapata (2008) to ensure that the target compressions are lexically and syntactically acceptable. To accomplish this purpose, we use the Oak system[3] (Sekine, 2002) and the Charniak parser[4] (Charniak, 1999) to obtain information regarding parts-of-speech and grammatical relations in a sentence.

## 2.2 Topic Signature Model

We use a topic signature modeling approach (Lin and Hovy, 2000) to identify the important content words from the original source sentence. The important words are considered to have significantly greater probability of occurring in a given text compared to that in a large background corpus. We incorporate this importance score into the objective function of the bigram language model (Section 2.1) to ensure that the target compression prefers to keep important content words. We use a topic signature computation tool[5] for this purpose. The background corpus that is used in this tool contains 5000 documents from the English GigaWord Corpus. Our modified objective function becomes:

$$Maximize \sum_i \delta_i \cdot I(w_i) + \sum_i a_i \cdot P(w_i|start) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} \cdot P(w_j|w_i) + \sum_i b_i \cdot P(end|w_i) \quad (12)$$

where $I(w_i)$ denotes the importance score of the $i$-th word.

## 2.3 Bigram Language Model with Semantic Constraints

Yoshikawa et al. (2012) have proposed a set of formulas called Markov Logic Network (MLN) to build a semantically motivated sentence compression model and showed that their model achieves improved performance. We recast their formulas as constraints of our ILP model and incorporate them into the bigram language model. The main idea is to utilize the predicate-argument relations of a sentence and define constraints based on semantic roles to improve the weaknesses of the lexical and syntactical constraints. In this manner, we can ensure that the target compression contains meaningful information. For this purpose, we parse the source sentence semantically using a Semantic Role Labeling (SRL) system (Kingsbury and Palmer, 2002; Hacioglu et al., 2003), ASSERT[6]. When presented with a sentence, ASSERT performs a full syntactic analysis of the sentence, automatically identifies all the verb predicates in that sentence, extracts features for all constituents in the parse tree relative to the predicate, and identifies and tags the constituents with the appropriate semantic arguments. We add the following additional constraints as the semantic constraints to our bigram language model (Section 2.1):

$$\delta_i = 1 \quad (13)$$
$$\forall i : w_i \; is \; a \; predicate$$

[3] http://nlp.cs.nyu.edu/oak/
[4] Available at ftp://ftp.cs.brown.edu/pub/nlparser/
[5] Available at http://www.cis.upenn.edu/ lannie/topicS.html
[6] Available at http://cemantix.org/assert.html

$$\delta_i - \delta_j = 0 \quad (14)$$
$$\forall i, j : w_j \text{ is an argument of predicate } w_i$$

$$\delta_i = 1 \quad (15)$$
$$\forall i : w_i \in [ARG0...ARG5]$$

$$\delta_i = 0 \quad (16)$$
$$\forall i : w_i \in \text{optional arguments}$$

Here, Constraint 13 guarantees that if a word is a predicate, it is included in the compression. Constraint 14 states that if a predicate is in compression, then its argument is also kept in the compression. In Constraint 15, we define that if a word denotes any of the possible semantic roles (i.e. *[ARG0...ARG5]* which are called *mandatory arguments*), it is included in the compression. On the other hand, we use Constraint 16 to restrict the inclusion of optional arguments[7] in the compression.

## 3 ILP for Query-focused Multi-document Summarization

The query-focused multi-document summarization inference problem can be formulated in terms of ILP. To represent the sentences included in the summary we define a set of indicator variables $\alpha_i$ that are set to 1 if $i$-th sentence is selected into the summary, and 0 otherwise. Let $Rel(i)$ be the relevance function that returns the relevance score of the $i$-th sentence. The score of a summary is the sum of the relevance scores of the sentences present in the summary. The inference task is solved by maximizing the overall score of a summary:

$$Maximize \sum_i Rel(i) * \alpha_i$$

such that $\forall i, j$ :

$$\alpha_i \in \{0, 1\} \quad (17)$$

$$Sim(i, j) * \left(\alpha_i + \alpha_j\right) \leq K \quad (18)$$

$$\sum_i Len(i) * \alpha_i \leq L \quad (19)$$

We incorporate three constraints into our formulation. The first constraint states that the variables are binary. The second constraint is the redundancy constraint that ensures that only one of the two similar sentences is chosen into the summary. $Sim(i, j)$ function returns a similarity score between the $i$-th and $j$-th sentences. Higher scores correspond to higher similarity between a pair of sentences. We assume a threshold $K$, that sets a tolerance limit to the acceptable similarity score between any two sentences. This value is empirically determined during experiments. The third constraint controls the length of the summary up to a maximum limit, $L$. $Len(i)$ denotes the length of the $i$-th sentence in words.

### 3.1 $Rel(i)$ Function

For each sentence, the $Rel(i)$ function returns a relevance score by combining a set of query-related and importance-oriented measures. The query-related measures calculate the similarity between each sentence and the given query while the importance-oriented measures denote the importance of a sentence in a given document (Chali and Hasan, 2012; Edmundson, 1969; Sekine and Nobata, 2001). For query-related measures, we consider n–gram overlap,

[7] There are some additional arguments or semantic roles that can be tagged by ASSERT. They are called *optional arguments* and they start with the prefix *ARGM*. These are defined by the annotation guidelines set in (Palmer et al., 2005).

longest common subsequence (LCS), weighted LCS, skip-bigram, exact word, synonym, hypernym/hyponym, gloss and basic elements (BE) overlap (Lin, 2004; Zhou et al., 2005) using WordNet (Fellbaum, 1998), and syntactic similarity (Collins and Duffy, 2001; Moschitti and Basili, 2006). To measure the importance of a sentence, we consider its position, length, similarity with topic title, and presence of certain named entities and cue words. The mean of these scores denote the relevance of a sentence.

#### 3.1.1 Query-related Measures

**n-gram Overlap** n–gram overlap measures the overlapping word sequences between the candidate document sentence and the query sentence (Lin, 2004).

**LCS** Given two sequences $S_1$ *and* $S_2$, the longest common subsequence (LCS) of $S_1$ *and* $S_2$ is a common subsequence with maximum length. We use this feature to calculate the longest common subsequence between a candidate sentence and the query.

**WLCS** Weighted Longest Common Subsequence (WLCS) improves the basic LCS method by remembering the length of consecutive matches encountered so far. Given two sentences X and Y, the WLCS score of X and Y can be computed using the similar dynamic programming procedure as stated in Lin (2004).

**Skip-Bigram** Skip-bigram measures the overlap of skip-bigrams between a candidate sentence and a query sentence. Skip-bigram counts all in-order matching word pairs while LCS only counts one longest common subsequence.

**Exact-word Overlap** This is a measure that counts the number of words matching exactly between the candidate sentence and the query sentence.

**Synonym Overlap** This is the overlap between the list of synonyms of the content words (i.e. nouns, verbs and adjectives) extracted from the candidate sentence and *query related words*[8].

**Hypernym/Hyponym Overlap** This is the overlap between the list of hypernyms (up to depth 2 in WordNet's hierarchy) and hyponyms (depth 3) of the nouns extracted from the sentence in consideration and *query related words*.

**Gloss Overlap** This is the overlap between the list of content words that are extracted from the gloss definition of the nouns in the sentence in consideration and *query related words*.

**Syntactic Feature** The syntactic similarity between the *query* and the *sentence* is calculated using a similar procedure discussed in Section 3.2.2, which gives the similarity score based on syntactic structures.

**Basic Element (BE) Overlap** We extract BEs (Hovy et al., 2006) for the sentences (or query) by using the BE package distributed by ISI[9]. We compute the Likelihood Ratio (LR) for each BE according to Zhou et al. (2005). We sort the BEs based on LR scores to produce a BE-ranked list. The ranked list contains important BEs at the top which may or may not be relevant to the

[8]To establish the query related words, we took a query and created a set of related queries by replacing its content words by their first-sense synonyms using WordNet.

[9]BE website:http://www.isi.edu/ cyl/BE

complex question. We filter out the BEs that are not related to the query and get the BE overlap score.

#### 3.1.2 Importance-oriented Measures

**Position of Sentences** Sentences that reside at the start and at the end of a document often tend to include the most valuable information. We manually inspected[10] the given document collection and found that the first and the last 3 sentences of a document often qualify to be considered for this feature. We assign the score 1 to them and 0 to the rest.

**Length of Sentences** Longer sentences contain more words and have a greater probability of containing valuable information. Therefore, a longer sentence has a better chance of inclusion in a summary[11]. We give the score 1 to a longer sentence and assign the score 0 otherwise. We manually investigated the document collection and set a threshold that a longer sentence should contain at least 11 words.

**Title Match** If we find a match such as exact word overlap, synonym overlap and hyponym overlap between the title and a sentence, we give it the score 1, otherwise 0.

**Named Entity** The score 1 is given to a sentence that contains a Named Entity class among: PERSON, LOCATION, ORGANIZATION, GPE (Geo-Political Entity), FACILITY, DATE, MONEY, PERCENT, TIME. We believe that the presence of a Named Entity increases the importance of a sentence. For example, the sentence *"Washington, D.C. is the capital of the United States"* has two named entities (i.e. locations) which denote that the sentence is important. We use the *OAK* System (Sekine, 2002), from New York University for Named Entity recognition.

**Cue Word Match** The probable relevance of a sentence is affected by the presence of pragmatic words such as "significant", "impossible", "in conclusion", "finally" etc. We use a cue word list of 228 words. We give the score 1 to a sentence having any of the cue words and 0 otherwise.

### 3.2 $Sim(i, j)$ Function

We employ four alternative redundancy constraints based on different sentence similarity functions (i.e. $Sim(i, j)$) using a) cosine similarity, b) syntactic similarity, c) semantic similarity, and d) extended string subsequence kernel (ESSK).

#### 3.2.1 Cosine Similarity Measure (COS)

The cosine similarity between the respective pair of sentences can be calculated by representing each sentence as a vector of term specific weights (Erkan and Radev, 2004). The term specific weights in the sentence vectors are products of local and global parameters. This is known as term frequency-inverse document frequency (tf-idf) model. The weight vector for a sentence $s$ is $\vec{v_s} = [w_{1,s}, w_{2,s}, \ldots, w_{N,s}]^T$, where,

$$w_{t,s} = tf_t \times \log \frac{|S|}{|\{t \in s\}|}$$

[10] We randomly investigated few newspaper articles and observed that sentences that reside at the start and at the end of a document often tend to include the most valuable information. The "Position of sentences" feature could be tuned to fit other genres of texts as well.

[11] The "Length of sentences" feature was exploited for summarization by extraction in general, which was our motivation to apply different compression models for the task.

Here, $tf_t$ is the term frequency (*tf*) of the term $t$ in a sentence $s$ (a local parameter). $\log \frac{|S|}{|\{t \in s\}|}$ is the inverse document frequency (idf) (a global parameter). $|S|$ is the total number of sentences in the corpus, and $|\{t \in s\}|$ is the number of sentences containing the term $t$.

### 3.2.2 Syntactic Similarity Measure (SYN)

Pasca and Harabagiu (2001) demonstrated that with the syntactic form one can see which words depend on other words. Syntactic features have been used successfully so far in *question answering* (Zhang and Lee, 2003; Moschitti et al., 2007; Moschitti and Basili, 2006). Inspired by the potential significance of using syntactic measures for finding similar texts, we get a strong motivation to use it as a redundancy measure in our optimization framework. The first step to calculate the syntactic similarity between two sentences is to parse the corresponding sentences into syntactic trees using the Charniak parser (Charniak, 1999). Once we build the syntactic trees, our next task is to measure the similarity between the trees. For this, every tree $T$ is represented by an $m$ dimensional vector $v(T) = \big(v_1(T), v_2(T), \cdots v_m(T)\big)$, where the i-th element $v_i(T)$ is the number of occurrences of the i-th tree fragment in tree $T$. The tree fragments of a tree are all of its sub-trees which include at least one production with the restriction that no production rules can be broken into incomplete parts. The tree kernel of two trees $T_1$ and $T_2$ is actually the inner product of $v(T_1)$ and $v(T_2)$ (Collins and Duffy, 2001):

$$TK(T_1, T_2) = v(T_1).v(T_2) \tag{20}$$

We define the indicator function $I_i(n)$ to be 1 if the sub-tree $i$ is seen rooted at node $n$ and 0 otherwise. It follows:

$$v_i(T_1) = \sum_{n_1 \in N_1} I_i(n_1)$$

$$v_i(T_2) = \sum_{n_2 \in N_2} I_i(n_2)$$

where, $N_1$ and $N_2$ are the set of nodes in $T_1$ and $T_2$ respectively. The *TK* (tree kernel) function gives the similarity score between a pair of sentences based on the syntactic structure.

### 3.2.3 Semantic Similarity Measure (SEM)

Shallow semantic representations can prevent the sparseness of deep structural approaches and the weakness of cosine similarity based models (Moschitti et al., 2007). As an example, PropBank (PB) (Kingsbury and Palmer, 2002) made it possible to design accurate automatic Semantic Role Labeling (SRL) systems (Hacioglu et al., 2003). Therefore, we get the feeling that an application of SRL as a redundancy measure might suit well, since the textual similarity between a pair of sentences relies on a deep understanding of the semantics of both. So, applying semantic similarity measurement as a $Sim(i, j)$ function is another noticeable contribution of this paper. To calculate the semantic similarity between two sentences, we first parse the corresponding sentences semantically using the Semantic Role Labeling (SRL) system, ASSERT. ASSERT is an automatic statistical semantic role tagger, that can annotate naturally occurring text with semantic arguments. We represent the annotated sentences using tree structures that are called semantic trees (ST). In the semantic tree, arguments are replaced with the most important word, often referred to as the semantic head. We look for noun, then verb, then adjective, then adverb to find the semantic head in the argument. If none of these is present, we take the first word of the argument as the semantic head. As in tree kernels (Section 3.2.2), common substructures cannot be composed by a node with only some of its

children as an effective ST representation would require, Moschitti et al. (2007) solved this problem by designing the Shallow Semantic Tree Kernel (SSTK) which allows to match portions of a ST. The SSTK function yields the similarity score between a pair of sentences based on their semantic structures.

#### 3.2.4 Extended String Subsequence Kernel (ESSK)

The ESSK is a simple extension of the Word Sequence Kernel (WSK) (Cancedda et al., 2003) and String Subsequence Kernel (SSK) (Lodhi et al., 2002) that can incorporate semantic information with the use of word senses. In original ESSK, each "alphabet" in SSK is replaced by a disjunction of an "alphabet" and its alternative (word senses) (Hirao et al., 2003). Here, all possible senses of a word are used as the alternatives. However, in our ESSK formulation, we consider each word in a sentence as an "alphabet", and the alternative as its disambiguated sense found through a dictionary based disambiguation approach. We use WordNet to find the semantic relations among the words in a text. We calculate the similarity score $\mathrm{Sim}(T_i, U_j)$ using ESSK where $T_i$ and $U_j$ are the two sentences. Formally, ESSK is defined as follows[12]:

$$K_{essk}(T,U) = \sum_{m=1}^{d} \sum_{t_i \in T} \sum_{u_j \in U} K_m(t_i, u_j)$$

$$K_m(t_i,u_j) = \begin{cases} val(t_i,u_j) & \text{if } m=1 \\ K'_{m-1}(t_i,u_j) \cdot val(t_i,u_j) & \end{cases}$$

Here, $K'_m(t_i,u_j)$ is defined below. $t_i$ and $u_j$ are the nodes of $T$ and $U$, respectively. The function $val(t,u)$ returns the number of attributes (i.e. words) common to the given nodes $t$ and $u$.

$$K'_m(t_i,u_j) = \begin{cases} 0 & \text{if } j=1 \\ \lambda K'_m(t_i,u_{j-1}) + K''_m(t_i,u_{j-1}) & \end{cases}$$

Here $\lambda$ is the decay parameter for the number of skipped words. We choose $\lambda = 0.5$ for this research. $K''_m(t_i,u_j)$ is defined as:

$$K''_m(t_i,u_j) = \begin{cases} 0 & \text{if } i=1 \\ \lambda K''_m(t_{i-1},u_j) + K_m(t_{i-1},u_j) & \end{cases}$$

Finally, the similarity measure is defined after normalization as below:

$$sim_{essk}(T,U) = \frac{K_{essk}(T,U)}{\sqrt{K_{essk}(T,T) K_{essk}(U,U)}}$$

## 4 Experiments

### 4.1 Task Description

We consider the query-focused multi-document summarization task defined in the Document Understanding Conference (DUC[13]), 2007. The task is: *"Given a complex question and a collection of relevant documents, the task is to synthesize a fluent, well-organized 250-word summary of the documents that answers the question(s) in the topic"*. We generate 250-word extract summaries for the topics of DUC-2007 using different combinations of sentence compression models (defined in Section 2) and alternative redundancy constraints (Section 3.2). DUC-2007 provided 45 document clusters each containing 25 news articles that came from the

[12]The formulae denotes a dynamic programming technique to compute the ESSK similarity score (Hirao et al., 2004) where $d$ is the vector space dimension i.e. the number of all possible subsequences of up to length $d$.

[13]http://duc.nist.gov/

AQUAINT corpus, which is comprised of newswire articles from the Associated Press and New York Times (1998-2000) and Xinhua News Agency (1996-2000). As we intend to study if the order of performing compression and extraction can affect the overall performance of the query-focused multi-document summarization, we compose three different models depending on the order to perform sentence compression and extraction: **(1) ComFirst:** In this approach, document sentences are compressed first (using different models as described in Section 2) and then the most relevant compressions are selected to form the summaries (according to Section 3), **(2) SumFirst:** In this approach, we extract the most important sentences first from the source documents (according to Section 3) and then compress them (using different models as described in Section 2) to form the summaries, and **(3) Combined:** Here, we perform compression and extraction jointly by combining the objective functions of Section 2 and Section 3 according to Martins and Smith (2009). Then we optimize the combined objective function to select a small number of most important sentences (from the source documents) whose compressions should be used to form a summary.

## 4.2 Solving the ILPs

To solve the proposed ILP formulations, we use *lp_solve*[14], a widely used Integer Linear Programming solver that implements Branch-and-Bound algorithm. For summarization, we solve an ILP for each topic in consideration and generate the corresponding query-focused summary. For a document cluster of average size (approximately 510 sentences), the solving process takes under 20 seconds on an Intel Pentium 4, 3.20 GHz desktop machine. For a larger document cluster (of size around 1000 sentences), it takes $90-120$ seconds to solve the ILP. For a smaller document set, the ILP is solved in a few seconds. For compression, we solve an ILP for each sentence in consideration. The solving process takes less than a second per sentence on average for all the compression models. For the joint extraction and compression model, we solve an ILP for each topic in consideration. The solving process is generally slower than solving the ILPs for only sentence extraction or compression as it takes $300-1200$ seconds depending on the document cluster size.

## 4.3 Evaluation Results and Discussion

### 4.3.1 Automatic Evaluation

The multiple "reference summaries" given by DUC-2007 are used in the evaluation of our summary content. We carried out the automatic evaluation of our summaries using the ROUGE (Lin, 2004) toolkit. Among different scores reported by ROUGE, unigram-based ROUGE score (ROUGE-1) has been shown to agree with human judgment most (Lin, 2003). We report the widely adopted important ROUGE metrics in the results: ROUGE-1 (unigram),and ROUGE-2 (bigram). The comparison between the systems in terms of their F-scores is given in Table 1. We also include the results of the official baseline systems, the best system (Pingali et al., 2007), and the average ROUGE scores of all the participating systems of DUC-2007. Baseline-1 returns all the leading sentences (up to 250 words) of the most recent document whereas baseline-2's main idea is to ignore the topic narrative while generating summaries using an HMM model[15].

The columns in Table 1 denote the use of alternative redundancy constraints in the optimization

[14] http://lpsolve.sourceforge.net/5.5/
[15] http://duc.nist.gov/pubs/2004papers/ida.conroy.ps

| | COS | | SYN | | SEM | | ESSK | | No Red. | | Comp. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | R1 | R2 | R1 | R2 | R1 | R2 | R1 | R2 | R1 | R2 | R1 | R2 |
| | | | | | | ComFirst | | | | | | |
| bi | 0.359 | 0.074 | 0.369 | 0.078 | 0.371 | 0.077 | 0.368 | 0.072 | 0.355 | 0.060 | | |
| topicS | 0.372 | 0.080 | 0.366 | 0.081 | 0.378 | 0.079 | 0.373 | 0.076 | 0.360 | 0.071 | | |
| bi+sem | 0.385 | 0.093 | 0.376 | 0.085 | 0.389 | 0.092 | 0.384 | 0.088 | 0.367 | 0.075 | | |
| | | | | | | SumFirst | | | | | | |
| bi | 0.368 | 0.076 | 0.365 | 0.079 | 0.388 | 0.096 | 0.370 | 0.088 | 0.362 | 0.071 | | |
| topicS | 0.374 | 0.083 | 0.371 | 0.084 | 0.392 | 0.101 | 0.378 | 0.091 | 0.365 | 0.074 | | |
| bi+sem | 0.388 | 0.096 | 0.382 | 0.091 | 0.405 | 0.113 | 0.391 | 0.101 | 0.374 | 0.083 | | |
| | | | | | | Combined | | | | | | |
| bi | 0.384 | 0.102 | 0.371 | 0.087 | 0.385 | 0.091 | 0.371 | 0.081 | 0.356 | 0.082 | | |
| topicS | 0.389 | 0.105 | 0.374 | 0.089 | 0.398 | 0.103 | 0.368 | 0.084 | 0.364 | 0.078 | | |
| bi+sem | 0.412 | 0.115 | 0.390 | 0.092 | 0.424 | 0.119 | 0.395 | 0.094 | 0.372 | 0.086 | | |
| No compr. | 0.400 | 0.108 | 0.399 | 0.109 | 0.412 | 0.111 | 0.396 | 0.105 | 0.381 | 0.091 | | |
| Baseline1 | | | | | | | | | | | 0.334 | 0.060 |
| Baseline2 | | | | | | | | | | | 0.400 | 0.093 |
| AverageDUC | | | | | | | | | | | 0.400 | 0.095 |
| Best System | | | | | | | | | | | 0.438 | 0.122 |

Table 1: Automatic Evaluation Results: Average ROUGE F-scores

framework whereas the rows stand for the use of different compression models[16]. From these results, we can clearly see the impact of using different sentence compression models on the overall summarization performance. In the **ComFirst** approach, we can see that the bigram model with semantic constraints outperforms all the other alternative models by a clear margin. We can also see the impact of different redundancy constraints on the overall performance. We observe that the use of semantic measure as the redundancy constraint yields the best performance. On the other hand, we see a clear improvement in almost all the scores when we follow the **SumFirst** approach. This phenomenon suggests that compressing the document sentences at the beginning often tend to reduce relevant information in the sentences for which we get lesser similarity matching when we calculate the relevance scores according to Section 3.1. In the **Combined** approach, we achieve better summarization performance than the other two approaches which denotes that the overall summary quality can be improved if a global optimization framework is utilized having a joint compression and extraction model. Again, we see that the bigram language model with semantic constraints along with the semantic redundancy constraint (used in the summarization model) yields the best performance. We also report the results of a *"No compression"* and a *"No redundancy"* baseline. Comparisons with these baselines also suggest that our bigram compression model with semantic constraints can improve the overall summarization performance if a **Combined** optimization framework is used in presence of **COS** or **SEM** redundancy constraints. These results also demonstrate that the absence of a redundancy constraint in the ILP framework for summarization really hurts the overall quality of the summaries. We also compare the scores of our model with the state-of-the-art systems of DUC-2007. From the results, we see that our semantically motivated models can mostly outperform the **DUC baselines** and the **AverageDUC** scores to show a clear improvement in the overall summarization performance while achieving a comparable performance with respect to the DUC-2007 best system. The differences between the models are computed

[16]The last few rows and columns are used to accommodate the scores of the baselines and the state-of-the-art systems.

to be statistically significant at $p < 0.05$ (using Student's t-test) except for the differences between **topicSig+SYN** and **bigram+SYN**, and **topicSig+ESSK** and **bigram+ESSK** in all the three approaches, between **topicSig+COS** and **bigram+COS** in the **Combined** approach, and between **"bigram+sem"+SEM** and **DUC Best System** in the **Combined** approach.

#### 4.3.2 Manual Evaluation

One of the important demerits of using sentence compression models is that they can degrade the linguistic quality of a summary by showing poor compression performance. ROUGE is not reliable to some researchers as there might be some linguistically bad summaries that get state-of-the-art ROUGE scores (Sjöbergh, 2007). So, we conduct an extensive manual evaluation in order to analyze the effectiveness of our approaches. Two self reported native English-speaking university graduate students judge the summaries for linguistic quality and overall responsiveness according to the DUC-2007 evaluation guidelines[17]. The given linguistic quality score is an integer between 1 (very poor) and 5 (very good) and is guided by consideration of the following factors: 1. Grammaticality, 2. Non-redundancy, 3. Referential clarity, 4. Focus, and 5. Structure and Coherence. The responsiveness score is also an integer between 1 (very poor) and 5 (very good) and is based on the amount of information in the summary that helps to satisfy the information need. The carried out user evaluation was subjective in nature specially while judging referential clarity, focus, coherence and overall responsiveness of the summaries. The inter-annotator agreement of Cohen's $\kappa = 0.43$ (Cohen, 1960) was computed that denotes a moderate degree of agreement (Landis and Koch, 1977) between the raters. Table 2 presents the average linguistic quality and overall responsive scores of all the systems. From these results, we can see that the use of different sentence compression models has a negative impact on the overall linguistic quality of the summaries. The reason behind this is that our bigram compression models were less aware of the underlying context in a sentence and hence, some word deletions resulted a loss in focus and coherence of the overall summaries. However, we observe that the semantically motivated models are showing an improved summarization performance; also, their overall responsiveness scores are comparable to the state-of-the-art systems. This suggests that the manual evaluation results are corresponding well to the automatic evaluation results. Considering the work of Gillick and Favre (2009) for a relative comparison, we find that both our automatic and manual evaluation results are corresponding fairly well to their results obtained on the TAC[18]-2008 data. Their ILP model with additional constraints to include sentence compression achieved an improvement in ROUGE-2 score over the "no compression" alternative while having reductions in manual evaluation scores. We perform a statistical significance test on our manual evaluation results at $p < 0.05$ using Student's t-test. The differences between the models are statistically significant except for the differences between **topicSig+COS** and **bigram+COS**, and **topicSig+SYN** and **bigram+SYN** in all the three approaches. The manual evaluation results also demonstrate that the use of different redundancy constraints certainly affects the overall performance of the proposed optimization framework for summarization[19]. From these experiments we can conclude that the semantic similarity measure can be used effectively as the $Sim(i, j)$ function to improve the performance of the traditional cosine similarity based approaches. We plan to make our created resources available to the scientific community.

[17] http://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

[18] Text Analysis Conference, http://www.nist.gov/tac/

[19] The selection of sentences in the optimal summaries varied due to different redundancy measures, hence, the linguistic quality scores also varied to reflect the differences in coherence, redundancy etc.

| | COS | | SYN | | SEM | | ESSK | | No Redundancy | | Comparison | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Models | LQ | Res. | LQ | Res. | LQ | Res. | LQ | Res. | LQ | Res. | LQ | Res. |
| ComFirst | | | | | | | | | | | | |
| bigram | 2.10 | 2.12 | 2.28 | 2.20 | 2.44 | 2.21 | 2.32 | 2.25 | 1.94 | 2.10 | | |
| topicSig | 2.14 | 2.30 | 2.45 | 2.27 | 2.48 | 2.78 | 2.39 | 2.46 | 2.08 | 2.26 | | |
| bigram+sem | 2.42 | 2.56 | 2.55 | 2.61 | 2.74 | 3.05 | 2.54 | 2.80 | 2.25 | 2.58 | | |
| SumFirst | | | | | | | | | | | | |
| bigram | 2.43 | 2.44 | 2.54 | 2.50 | 2.60 | 2.45 | 2.25 | 2.34 | 2.16 | 2.56 | | |
| topicSig | 2.48 | 2.56 | 2.65 | 2.69 | 2.72 | 2.66 | 2.48 | 2.55 | 2.27 | 2.68 | | |
| bigram+sem | 2.61 | 2.76 | 2.88 | 2.78 | 3.20 | 3.56 | 2.75 | 2.93 | 2.42 | 2.62 | | |
| Combined | | | | | | | | | | | | |
| bigram | 2.54 | 2.62 | 2.52 | 2.31 | 2.76 | 2.55 | 2.36 | 2.50 | 1.98 | 2.20 | | |
| topicSig | 2.62 | 2.75 | 2.68 | 2.38 | 2.80 | 2.62 | 2.45 | 2.64 | 2.14 | 2.31 | | |
| bigram+sem | 2.85 | 3.08 | 2.91 | 2.93 | 3.18 | 3.61 | 2.77 | 2.88 | 2.32 | 2.42 | | |
| No compression | 3.30 | 3.38 | 3.42 | 3.15 | 3.64 | 3.50 | 3.38 | 3.21 | 2.28 | 2.15 | | |
| Baseline1 | | | | | | | | | | | 4.24 | 1.86 |
| Baseline2 | | | | | | | | | | | 4.48 | 2.71 |
| Best System | | | | | | | | | | | 4.11 | 3.40 |

Table 2: Average linguistic quality (LQ) and responsiveness scores (Res.)

## Conclusion and Future Work

We have analyzed the effectiveness of using different ILP-based sentence compression models for the task of query-focused multi-document summarization. Our empirical evaluation suggested that the semantically motivated sentence compression models can enhance the overall summarization performance in presence of the semantic redundancy constraint in the summarization model and this can be achieved irrespective of the compression and extraction order followed during the process. Our results also demonstrated that a combined optimization framework of compression and extraction can achieve better performance than the other two considered approaches effectively. We also found that the *SumFirst* approach shows superior performance to that of the *ComFirst* approach suggesting the fact that extracting the most important sentences before compression is a more effective way of summarization. We have also used different textual similarity measurement techniques as the redundancy constraints of the ILP-based summarization framework and performed an extensive experimental evaluation to show their impact on the overall summarization performance. Experimental results showed that the use of semantic similarity measure as the $Sim(i, j)$ function in the redundancy constraint yields the best performance. Overall, our global optimization frameworks showed promising performance with respect to the state-of-the-art systems. We look forward to apply our approach to other available datasets of DUC-2005 and DUC-2006. The findings should hold for these datasets as well as for other genres of datasets since we believe that our ILP-based compression and summarization models could be tuned to fit them. We also plan to use other automatic measures (Saggion et al., 2010; Pitler et al., 2010) to evaluate our approach.

## Acknowledgments

The research reported in this paper was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada – discovery grant and the University of Lethbridge.

## References


Berg-Kirkpatrick, T., Gillick, D., and Klein, D. (2011). Jointly Learning to Extract and Compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 481–490. ACL.

Bouayad-Agha, N., Gil, A., Valentin, O., and Pascual, V. (2006). A Sentence Compression Module for Machine-Assisted Subtitling. In *Computational Linguistics and Intelligent Text Processing*, pages 490–501. Springer Berlin Heidelberg.

Cancedda, N., Gaussier, E., Goutte, C., and Renders, J. M. (2003). Word Sequence Kernels. *Journal of Machine Learning Research*, 3:1059–1082.

Carbonell, J. and Goldstein, J. (1998). The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 335–336, Melbourne, Australia.

Chali, Y. and Hasan, S. A. (2012). Query-focused Multi-document Summarization: Automatic Data Annotations and Supervised Learning Approaches. *Natural Language Engineering*, 18(1):109–145.

Charniak, E. (1999). A Maximum-Entropy-Inspired Parser. In *Technical Report CS-99-12*, Brown University, Computer Science Department.

Clarke, J. and Lapata, M. (2008). Global Inference for Sentence Compression An Integer Linear Programming Approach. *Journal of Artificial Intelligence Research*, 31(1):399–429.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Cohn, T. and Lapata, M. (2008). Sentence Compression Beyond Word Deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK.

Collins, M. and Duffy, N. (2001). Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems*, pages 625–632, Vancouver, Canada.

Daumé III, H. and Marcu, D. (2005). Bayesian Multi-Document Summarization at MSE. In *Proceedings of the Workshop on Multilingual Summarization Evaluation (MSE)*.

Denis, P. and Baldridge, J. (2007). Joint Determination of Anaphoricity and Coreference Resolution using Integer Programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics;*, pages 236–243. ACL.

Edmundson, H. P. (1969). New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery (ACM)*, 16(2):264–285.

Erkan, G. and Radev, D. R. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Fellbaum, C. (1998). WordNet - An Electronic Lexical Database. Cambridge, MA. MIT Press.

Filippova, K. (2010). Multi-Sentence Compression: Finding Shortest Paths in Word Graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 322–330. ACL.

Galanis, D. and Androutsopoulos, I. (2010). An Extractive Supervised Two-Stage Method for Sentence Compression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 885–893. ACL.

Gillick, D. and Favre, B. (2009). A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 10–18. ACL.

Goldstein, J., Kantrowitz, M., Mittal, V., and Carbonell, J. (1999). Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In *Proceedings of the 22nd International ACM Conference on Research and Development in Information Retrieval, SIGIR*, pages 121–128, Berkeley, CA.

Hacioglu, K., Pradhan, S., Ward, W., Martin, J. H., and Jurafsky, D. (2003). Shallow Semantic Parsing Using Support Vector Machines. In *Technical Report TR-CSLR-2003-03*, University of Colorado.

Hirao, T., , Suzuki, J., Isozaki, H., and Maeda, E. (2004). Dependency-based Sentence Alignment for Multiple Document Summarization. In *Proceedings of COLING 2004*, pages 446–452, Geneva, Switzerland. COLING.

Hirao, T., Suzuki, J., Isozaki, H., and Maeda, E. (2003). NTT's Multiple Document Summarization System for DUC2003. In *Proceedings of the Document Understanding Conference*.

Hovy, E., Lin, C. Y., Zhou, L., and Fukumoto, J. (2006). Automated Summarization Evaluation with Basic Elements. In *Proceedings of the Fifth Conference on Language Resources and Evaluation*, Genoa, Italy.

Jing, H. (2000). Sentence Reduction for Automatic Text Summarization. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 310–315. ACL.

Kingsbury, P. and Palmer, M. (2002). From Treebank to PropBank. In *Proceedings of the International Conference on Language Resources and Evaluation*, Las Palmas, Spain.

Knight, K. and Marcu, D. (2000). Statistics-Based Summarization - Step One: Sentence Compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 703–710. AAAI Press.

Knight, K. and Marcu, D. (2002). Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. *Artificial Intelligence*, 139(1):91–107.

Kupiec, J., Pedersen, J., and Chen, F. (1995). A Trainable Document Summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1995)*, pages 68–73, Seattle, Washington, USA.

Landis, J. R. and Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174.

Lin, C. Y. (2003). Improving Summarization Performance by Sentence compression: A Pilot Study. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages - Volume 11*, pages 1–8. ACL.

Lin, C. Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics*, pages 74–81, Barcelona, Spain.

Lin, C.-Y. and Hovy, E. H. (2000). The Automated Acquisition of Topic Signatures for Text Summarization. In *Proceedings of the 18th conference on Computational linguistics*, pages 495–501.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text Classification using String Kernels. *Journal of Machine Learning Research*, 2:419–444.

Madnani, N., Zajic, D., Dorr, B., Ayan, N. F., and Lin, J. (2007). Multiple Alternative Sentence Compressions for Automatic Text Summarization. In *In Proceedings of the 2007 Document Understanding Conference (DUC-2007) at NLT/NAACL 2007*.

Mani, I. and Maybury, M. (1999). *Advances in Automatic Text Summarization*. MIT Press.

Martins, A. F. T. and Smith, N. A. (2009). Summarization with a Joint Model for Sentence Extraction and Compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 1–9. ACL.

Mcdonald, R. (2006). Discriminative Sentence Compression with Soft Syntactic Constraints. In *In Proceedings of the 11th Conference of the EACL*.

McDonald, R. (2007). A Study of Global Inference Algorithms in Multi-document Summarization. In *Proceedings of the 29th European conference on IR research*, ECIR'07, pages 557–564. Springer-Verlag.

Molina, A., Torres-Moreno, J., SanJuan, E., da Cunha, I., Sierra, G., and Velázquez-Morales, P. (2011). Discourse Segmentation for Sentence Compression. In *Proceedings of the 10th Mexican international conference on Advances in Artificial Intelligence - Volume Part I*, pages 316–327. Springer-Verlag.

Moschitti, A. and Basili, R. (2006). A Tree Kernel Approach to Question and Answer Classification in Question Answering Systems. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, Italy.

Moschitti, A., Quarteroni, S., Basili, R., and Manandhar, S. (2007). Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classificaion. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 776–783, Prague, Czech Republic. ACL.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31:71–106.

Pasca, M. and Harabagiu, S. M. (2001). Answer Mining from On-Line Documents. In *Proceedings of the Association for Computational Linguistics 39th Annual Meeting and 10th Conference of the European Chapter Workshop on Open-Domain Question Answering*, pages 38–45, Toulouse, France.

Pingali, P., K., R., and Varma, V. (2007). IIIT Hyderabad at DUC 2007. In *Proceedings of the Document Understanding Conference*, Rochester. NIST.

Pitler, E., Louis, A., and Nenkova, A. (2010). Automatic Evaluation of Linguistic Quality in Multi-document Summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 544–554. ACL.

Punyakanok, V., Roth, D., Yih, W., and Zimak, D. (2004). Semantic Role Labeling via Integer Linear Programming Inference. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04. ACL.

Riedel, S. and Clarke, J. (2006). Incremental Integer Linear Programming for Non-projective Dependency Parsing. In *In EMNLP*, pages 129–137.

Roth, D. and Yih, W. (2004). A Linear Programming Formulation for Global Inference in Natural Language Tasks. In *In Proceedings of CoNLL-2004*, pages 1–8.

Saggion, H., Torres-Moreno, J., Cunha, I., and SanJuan, E. (2010). Multilingual Summarization Evaluation without Human Models. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1059–1067. ACL.

Sekine, S. (2002). Proteus Project OAK System (English Sentence Analyzer), http://nlp.nyu.edu/oak.

Sekine, S. and Nobata, C. A. (2001). Sentence Extraction with Information Extraction Technique. In *Proceedings of the Document Understanding Conference (DUC 2001)*, New Orleans, Louisiana, USA.

Sjöbergh, J. (2007). Older Versions of the ROUGEeval Summarization Evaluation System Were Easier to Fool. *Information Processing and Management*, 43:1500–1505.

Yoshikawa, K., Iida, R., Hirao, T., and Okumura, M. (2012). Sentence Compression with Semantic Role Constraints. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 349–353, Jeju Island, Korea. ACL.

Zajic, D., Dorr, B. J., Lin, J., and Schwartz, R. (2007). Multi-candidate Reduction: Sentence Compression as a Tool for Document Summarization Tasks. *Information Processing and Management*, 43(6):1549–1570.

Zhang, A. and Lee, W. (2003). Question Classification using Support Vector Machines. In *Proceedings of the Special Interest Group on Information Retrieval*, pages 26–32, Toronto, Canada. ACM.

Zhou, L., Lin, C. Y., and Hovy, E. (2005). A BE-based Multi-dccument Summarizer with Query Interpretation. In *Proceedings of Document Understanding Conference*, Vancouver, B.C., Canada.