Using External Resources and Joint Learning for Bigram Weighting in ILP-Based Multi-Document Summarization

Chen Li[1], **Yang Liu**[1], **Lin Zhao**[2]
[1] Computer Science Department, The University of Texas at Dallas
Richardson, Texas 75080, USA
[2] Research and Technology Center, Robert Bosch LLC
Palo Alto, California 94304, USA
{chenli,yangl@hlt.utdallas.edu}
{lin.zhao@us.bosch.com}

Abstract

Some state-of-the-art summarization systems use integer linear programming (ILP) based methods that aim to maximize the important concepts covered in the summary. These concepts are often obtained by selecting bigrams from the documents. In this paper, we improve such bigram based ILP summarization methods from different aspects. First we use syntactic information to select more important bigrams. Second, to estimate the importance of the bigrams, in addition to the internal features based on the test documents (e.g., document frequency, bigram positions), we propose to extract features by leveraging multiple external resources (such as word embedding from additional corpus, Wikipedia, Dbpedia, WordNet, SentiWordNet). The bigram weights are then trained discriminatively in a joint learning model that predicts the bigram weights and selects the summary sentences in the ILP framework at the same time. We demonstrate that our system consistently outperforms the prior ILP method on different TAC data sets, and performs competitively compared to other previously reported best results. We also conducted various analyses to show the contributions of different components.

1 Introduction

Extractive summarization is a sentence selection problem: identifying important summary sentences from one or multiple documents. Many methods have been developed for this problem, including supervised approaches that use a classifier to predict whether or not a sentence is in the summary, or unsupervised methods such as graph-based approaches to rank the sentences. Recently global optimization methods such as integer linear programming (ILP) have been shown to be quite powerful for this task. For example, Gillick et al. (2009) used ILP to achieve the best result in the TAC 09 summarization task. The core idea of this summarization method is to select the summary sentences by maximizing the sum of the weights of the language concepts that appear in the summary. Bigrams are often used as these language concepts because Gillick et al. (2009) stated that the bigrams gave consistently better performance than unigrams or trigrams for a variety of ROUGE measures. The association between the language concepts and sentences serves as the constraints. This ILP method is formally represented as below (see (Gillick et al., 2009) for more details):

$$\max \quad \sum_i w_i c_i \tag{1}$$
$$s.t. \quad s_j Occ_{ij} \leq c_i \tag{2}$$
$$\sum_j s_j Occ_{ij} \geq c_i \tag{3}$$
$$\sum_j l_j s_j \leq L \tag{4}$$
$$c_i \in \{0,1\}\ \forall i \tag{5}$$
$$s_j \in \{0,1\}\ \forall j \tag{6}$$

c_i and s_j are binary variables that indicate the presence of a concept and a sentence respectively. l_j is the sentence length and L is maximum length of the generated summary. w_i is a concept's weight and Occ_{ij} means the occurrence of concept i in sentence j. Inequalities (2)(3) associate the sentences

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 778–787,
Denver, Colorado, May 31 – June 5, 2015.

and concepts. They ensure that selecting a sentence leads to the selection of all the concepts it contains, and selecting a concept only happens when it is present in at least one of the selected sentences.

In such ILP-based summarization methods, how to determine the concepts and measure their weights is the key factor impacting the system performance. Intuitively, if we can successfully identify the important key bigrams to use in the ILP system, or assign large weights to those important bigrams, the system generated summary sentences will contain as many important bigrams as possible. The oracle experiment in (Gillick et al., 2008) showed that if they just use the bigrams extracted from human generated summaries as the input of the ILP system, much better ROUGE scores can be obtained than using the automatically selected bigrams.

In this paper, we adopt the ILP summarization framework, but make improvement from three aspects. First, we use the part-of-speech tag and constituent parse information to identify important bigram candidates: bigrams from base NP (noun phrases) and bigrams containing verbs or adjectives. This bigram selection method allows us to keep the important bigrams and filter useless ones. Second, to estimate the bigrams' weights, in addition to using information from the test documents, such as document frequency, syntactic role in a sentence, etc., we utilize a variety of external resources, including a corpus of news articles with human generated summaries, Wiki documents, description of name entities from DBpedia, WordNet, and SentiWordNet. Discriminative features are computed based on these external resources with the goal to better represent the importance of a bigram and its semantic similarity with the given query. Finally, we propose to use a joint bigram weighting and sentence selection process to train the feature weights. Our experimental results on multiple TAC data sets show the competitiveness of our proposed methods.

2 Related Work

Optimization methods have been widely used in extractive summarization lately. McDonald (2007) first introduced the sentence level ILP for summarization. Later Gillick et al. (2009) revised it to concept-based ILP, which is similar to the Budgeted Maximal Coverage problem in (Khuller et al., 1999). The concept-based ILP system performed very well in the TAC 2008 and 2009 summarization task (Gillick et al., 2008; Gillick et al., 2009). After that, the global optimization strategy attracted increasing attention in the summarization task. Lin and Bilmes (2010) treated the summarization task as a maximization problem of submodular functions. Davis et al. (2012) proposed an optimal combinatorial covering algorithm combined with LSA to measure the term weight for extractive summarization. Takamura and Okumura (2009) also defined the summarization problem as a maximum coverage problem and used a branch-and-bound method to search for the optimal solution. Li et al. (2013b) used the same ILP framework as (Gillick et al., 2009), but incorporated a supervised model to estimate the bigram frequency in the final summary.

Similar optimization methods are also widely used in the abstractive summarization task. Martins and Smith (2009) leveraged ILP technique to jointly select and compress sentences for multi-document summarization. A novel summary guided sentence compression was proposed by (Li et al., 2013a) and it successfully improved the summarization performance. Woodsend and Lapata (2012) and Li et al. (2014) both leveraged constituent parser trees to help sentence compression, which is also modeled in the optimization framework. But these kinds of work involve using complex linguistic information, often based on syntactic analysis.

Since the language concepts (or bigrams) can be considered as key phrases of the documents, the other line related to our work is how to extract and measure the importance of key phrases from documents. In particular, our work is related to key phrase extraction by using external resources. A survey by (Hasan and Ng, 2014) showed that using external resources to extract and measure key phrases is very effective. In (Medelyan et al., 2009), Wikipedia-based key phrases are determined based on a candidate's document frequency multiplied by the ratio of the number of Wikipedia articles containing the candidate as a link to the number of articles containing the candidate. Query logs were also used as another external resource by (Yih et al., 2006) to exploit the observation that a candidate is potentially important if it was used as a search

query. Similarly terminological databases have been exploited to encode the salience of candidate key phrases in scientific papers (Lopez and Romary, 2010). In summarization, external information has also been used to measure word salience. Some TAC systems like (Kumar et al., 2010; Jia et al., 2010) used Wiki as an important external resource to measure the words' importance, which helped improve the summarization results. Hong and Nenkova (2014) introduced a supervised model for predicting word importance that incorporated a rich set of features. Tweets information is leveraged by (Wei and Gao, 2014) to help generate news highlights.

In this paper our focus is on choosing useful bigrams and estimating accurate weights to use in the concept-based ILP methods. We explore many external resources to extract features for bigram candidates, and more importantly, propose to estimate the feature weights in a joint process via structured perceptron learning that optimizes summary sentence selection.

3 Summarization System

In this study we use the ILP-based summarization framework (Formulas 1-6) that tries to maximize the weights of the selected concepts (bigrams) under the summary length constraint. Our focus is on better selection of the bigrams and estimation of the bigram weights. We use syntax tree and POS of tokens to help filter some useless bigrams. Then supervised methods are applied to predict the bigram weights. The rich set of features we use is introduced in Section 4. In the following we describe how to select important bigrams and how the feature weights are trained.

3.1 Bigram Selection

In (Gillick et al., 2009), bigrams whose document frequency is higher than a predefined threshold (df=3 in previous work) are used as the concepts in the ILP model. The weight for these bigrams in the ILP optimization objective function (Formula 1) is simply set as their document frequency. Although this setting has been demonstrated to be quite effective, its gap with the oracle experiment (using bigrams that appear in the human summaries) is still very large, suggesting potential gains by using better bigrams/concepts in the ILP optimization method. Details are described in (Gillick et al., 2009).

In this paper, rather than considering all the bigrams, we propose to utilize syntactic information to help select important bigrams. Intuitively bigrams containing content words carry more topic related information. As proven in (Klavans and Kan, 1998), nouns, verbs, and adjectives were indeed beneficial in document analysis. Therefore we focus on choosing bigrams containing these words. First, we use a bottom-up strategy to go through the constituent parse tree and identify the 'NP' nodes in the lowest level of the tree. Then all the bigrams in these base NPs are kept as candidates. Second, we find the verbs and adjectives from the sentence based on the POS tags, and construct bigrams by concatenating the previous or the next word of that verb or adjective. If these bigrams are not included in those already found from the base NPs, they are added to the bigram candidates. After the above filtering, we further drop bigrams if both words are stop words, as previous work in (Gillick et al., 2009).

3.2 Weight Training

We propose to train the feature weights in a joint learning fashion. In the ILP summarization framework, we use the following new objective function:

$$\max \quad \sum_i (\theta \cdot \mathbf{f}(b_i)) c_i \tag{7}$$

We replace the w_i in Formula 1 with a vector inner product of bigram features and their corresponding weights. Constraints remain the same as those in Formula 2 to 6.

To train the model (feature weights), we leverage structured perceptron strategy (Collins, 2002) to update the feature weights whenever the hypothesis offered by the ILP decoding process is incorrect. Binary class labels are used for bigrams in the learning process, that is, we only consider whether a bigram is in the system generated summary or human summaries, not their term or document frequency. During perceptron training, a fixed learning rate is used and parameters are averaged to prevent overfitting.

4 Features for Bigrams

We use a rich set of features to represent each bigram candidate, including internal features based on

the test documents, and features extracted from external resources. The goal is to better predict the importance of a bigram, which we expect will help the ILP module better determine whether to include the bigram in the summary.

4.1 Internal Features

These features are generated from the provided test documents (note our task is multi-document summarization, and there is a given query topic. See Section 5 for the description of tasks and data).

- Frequency of the bigram in the entire set.
- Frequency of the bigram in related sentences.[1]
- Document frequency of the bigram in the entire set.
- Is this bigram in the first 1/2/3 sentence?
- Is this bigram in the last 1/2/3 sentence?
- Similarity with the topic title, calculated by the number of common tokens in these two strings, divided by the length of the longer string.

4.2 Importance Score based on Language Models

The idea is to train two language models (LMs), one from the original documents, and the other one from the summaries, and compare the likelihood of a bigram generated by these two LMs, which can indicate how often a bigram is used in a summary. Similar to previous work in (Hong and Nenkova, 2014), we leveraged The New York Times Annotated Corpus (LDC Catalog No: LDC2008T19), which has the original news articles and human generated abstracts. We build two language models, from the news articles and the corresponding summaries respectively. We used about 160K abstract-original pairs. The KL scores for a bigram are defined as follows:

$$KL(LM_A|LM_O)(b) = Pr_A(b) * ln\frac{Pr_A(b)}{Pr_O(b)} \quad (8)$$

$$KL(LM_O|LM_A)(b) = Pr_O(b) * ln\frac{Pr_O(b)}{Pr_A(b)} \quad (9)$$

[1]Note that we do not use all the sentences in the ILP module. The 'relevant' sentences are those that have at least one bigram with document frequency larger than or equal to three.

where (LM_A) and (LM_O) are the LMs from the abstracts and the original news articles. Note that one difference from (Hong and Nenkova, 2014) is that we calculate these scores for a bigram, not a word. As (Hong and Nenkova, 2014) showed, a higher value from the score in Formula 8 means the words are favored in the summaries, and vice verse in Formula 9. In addition to the above features, we also include the likelihood $Pr_A(b)$ and $Pr_O(b)$ based on the two LMs, and the absolute and relative difference between them: $Pr_A(b) - Pr_O(b), Pr_A(b)/Pr_O(b)$.

4.3 Similarity based on Word Embedding Representation

Given the recent success of the continuous representation for words, we propose to use an unsupervised method to induce dense real-valued low dimensional word embedding, and then use the inner product as a measure of semantic similarity between two strings. In the word embedding model, every word can be represented by a vector $\vec{w}$. We define the similarity between two sequences $S1 = x_1, x_2, ...x_k$ and sequence $S2 = y_1, y_2, ...y_l$ as the average pairwise similarity between any two words in them:

$$Sim(S1, S2) = \frac{\sum_{i=1}^{k}\sum_{j=1}^{l}\vec{x}_i \cdot \vec{y}_j}{k * l} \quad (10)$$

Based on such word embedding models, we derive two similarity features: (1) similarity between a bigram and the topic query, and (2) similarity between a bigram and top-k most frequent unigrams in this topic. We trained two word embedding models, from the abstract and news article collections in the New York Times Annotated Corpus, and thus have two sets of the above similarity features. We use the continuous bag-of-words model introduced by (Mikolov et al., 2013), and the tool word2vec[2] to obtain the word embeddings.

4.4 Similarity based on WordNet[3]

Similar to the above method, here we still focus on measuring the similarity between a bigram and the topic query, but based on WordNet. We use WordNet to identify the synonyms of nouns, verbs,

[2]https://code.google.com/p/word2vec/

[3]http://wordnet.princeton.edu/

and adjectives from each bigram and the query of the topic. Then every bigram and sentence can be represented as a bag of synonyms of the original words. Finally based on these synonyms we leverage the following four similarity measurements: Lin Similarity (Lin, 1998), Wu-Palmer Similarity (Wu and Palmer, 1994), Jiang-Conrath Similarity (Jiang and Conrath, 1997), and Resnik Similarity (Resnik, 1995). These four similarity measurements are all implemented in the NLTK toolkit[4]. We expect that these features would improve the estimation accuracy because they can overcome the ambiguity and the diversity of the vocabulary.

4.5 Importance based on Wikipedia

Wikipedia is a very popular resource used in many different tasks. In order to obtain more precise external information from Wikipedia for our task, we collect the articles from Wikipedia by two steps. If the query is already the title of a wiki page, we will not further gather other wiki pages for this topic. Otherwise, we first search for the wiki pages for the given topic query and description (if available) using Google advanced search function to find pages from http://en.wikipedia.org/. For each returned wiki page, we further calculate its similarity between its abstract and the test documents' top k frequent words. We select 3 most similar pages as the external Wiki resource for this topic. For these wikipages, we split into two parts: abstract and content.[5] The features are the following: For each bigram, we collect its tf*idf score from the abstract and content part respectively, and the average tf*idf value of the unigrams in the bigram candidate. In addition, we design two boolean features that represent whether a bigram is the top-k most frequent ones in the abstract or the content part of the Wikepages.

4.6 DBpedia[6] for Extending Name Entity

DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and its Spotlight Service[7] is an entity linking tool to connect free text to DBpedia through the recognition and disambiguation of entities and concepts from the DBpedia Knowledge Base. We use this service to extract the entity from each sentence, and if the recognized entity is also identified as a named entity by Stanford CoreNLP[8], we use this entity's DBpedia abstract content to extend the bigrams. For example, in the bigram 'Kashmir area', the word 'Kashmir' is recognized as an entity by both (Stanford CoreNLP and DBpedia Spotlight service), then we use the description for 'Kashmir' from DBpedia[9] to extend this bigram, and calculate the cosine similarity between this description and the topic query and top-k most frequent unigrams in the documents.

4.7 Sentiment Feature from SentiWordNet[10]

SentiWordNet (Baccianella et al., 2010) is an extension on WordNet and it further assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity. The sentiment score of a bigram is the average score of the two words in the bigram.

To sum up, the features we use include the internal features, and external ones derived from various resources: news article corpus with summaries, Wikipeida, DBpedia, WordNet and SentiWordNet. Some external features represent the inherent importance of bigrams. For example, features extracted from the news article corpus and wikipedia are used to represent how often bigrams are used in summary/abstract compared to the entire document. Some external features are used to better compute semantic similarity, for example, features from the word embedding methods, DBpedia, and WordNet.

5 Experiments

5.1 Data and Experiment Setup

We evaluate our methods using several recent TAC data sets, from 2008 to 2011. The TAC summarization task is to generate at most 100 words summaries from 10 documents for a given topic query

[4]http://www.nltk.org/

[5]Every Wikipage has a table of contents. The part before that is considered as abstract and the part after that is the content of that page.

[6]http://dbpedia.org/About

[7]http://blog.dbpedia.org/2014/07/21/dbpedia-spotlight-v07-released/

[8]http://nlp.stanford.edu/software/corenlp.shtml

[9]The Indian subcontinent is a southerly region of Asia, mostly situated on the Indian Plate and projecting southward into the Indian Ocean. Definitions of the extent of the Indian subcontinent differ but it usually includes the core lands of India, Pakistan, and Bangladesh

[10]http://sentiwordnet.isti.cnr.it/

consisting of a title and more detailed description (this is unavailable in 2010 and 2011 data). When evaluating on one TAC data set, we use the data from the other three years as the training set. All the summaries are evaluated using ROUGE (Lin, 2004; Owczarzak et al., 2012). In all of our experiments, we use Stanford CoreNLP toolkit to tokenize the sentences, extract name entities and POS tags. Berkeley Parser (Petrov et al., 2006) is used to get the constituent parse tree for every sentence. An academic free solver [11] does all the ILP decoding.

5.2 Results and Analysis

5.2.1 Summarization Results

Table 1 shows the ROUGE-2 results of our proposed joint system, the ICSI system (which uses document frequency threshold to select bigram concepts and uses df as weights), the best performing system in the NIST TAC evaluation, and the state of the art performance we could find. The result of our proposed method is statistically significantly better than that of ICSI ILP ($p < 0.05$ based on paired t-test). It is also statistically significantly ($p < 0.05$) better than that of TAC Rank1 except 2011, and previous best in 2008 and 2010. The 2011 previous best results from (Ng et al., 2012) involve some rule-based sentence compression, which improves the ROUGE value. If we apply the same or similar rule-based sentence compression on our results, and the ROUGE-2 of our proposed method improves to 14.38.

	2008	2009	2010	2011
ICSI ILP	10.23	11.60	10.03	12.71
TAC Rank1	10.38	12.16	9.57	13.44
Previous Best	10.76†	12.46†	10.8‡	13.93∗
Proposed Method	**11.84**	**12.77**	**11.78**	**13.97**

Table 1: ROUGE-2 summarization results.† is from (Li et al., 2013b), ‡ is from (Davis et al., 2012), and ∗ is from (Ng et al., 2012).

5.2.2 The Effect of Bigram Selection

In our experiments, the document frequency threshold used to filter the bigrams is 3, the same as that in (Gillick et al., 2009), in order to make a better comparison with previous work. Figure 1 shows the percentage of the correct bigrams (those in the human reference summaries) by our proposed selection method and the original ICSI system which just used document frequency based selection. We can see that our selection method yields a higher percent of the correctly chosen bigrams. Since our proposed method is slightly aggressive when filtering bigrams, the absolute number of the correct bigrams decreased. However, our filtering method successfully removes more useless bigrams, resulting in a higher percentage of the correct bigrams.

Table 2 shows the summarization results when using different bigrams: the method used in the ICSI ILP system, that is, document frequency based selection/filtering and our selection method. Both of them use document frequency as the bigram weight in the ILP summarization module. The results show that just by changing the input bigrams, our method has already outperformed the ICSI system, which means the selection of bigram indeed has an impact on summarization results.

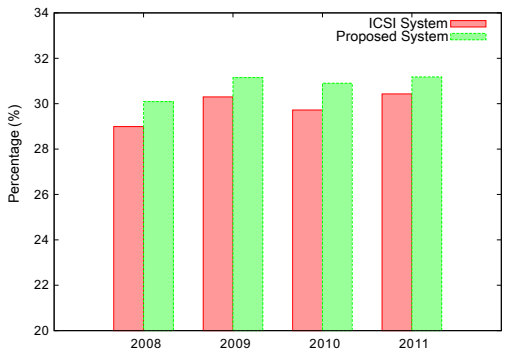

Figure 1: Percentage of correct bigrams in the selected bigrams from ICSI and our proposed system.

	2008	2009	2010	2011
ICSI ILP	10.23	11.60	10.03	12.71
Ours	10.26	11.65	10.25	12.75

Table 2: ROUGE-2 summarization results when using different bigrams, both using document frequencies as weights.

[11] http://www.gurobi.com

5.2.3 The Effect of Features

Next we evaluate the contributions of different features. We show results for four experiments: (i) use just one type of features; (ii) combine the internal features with features from just one external resource; (iii) incrementally add external resources one by one; (iv) leave out each feature type.

Table 3 shows the ROUGE-2 results when we only apply one type of features. First, we can see that the system with the internal features has already outperformed the baseline which used document frequency as the weight. It shows that the other chosen internal features (beyond document frequency) are useful. Second, when we use the features from only one external resource, the results from some resources are competitive compared to that from the system using internal features. In particular, when using the LM scores, Wiki or Word Embedding features, the results are slightly better than the internal features. Using DBpedia or SentiWordNet has worse results than the internal features. This is because the SentiWordNet features themselves are not very discriminative. For DBpedia, since it only has feature values for the bigrams containing name entities, it will only assign weights for those bigrams. Therefore, only considering DBpedia features means that the ILP decoder would prefer to choose bigrams that are name entities with positive weights.

	2008	2009	2010	2011
Internal	10.40	11.76	10.42	12.91
LM	10.58	11.86	10.48	12.94
Word Embedding	10.67	11.96	10.58	13.02
Wikipedia	10.61	11.90	10.52	13.00
DBpedia	8.35	9.85	9.46	11.00
WordNet	10.39	11.76	10.40	12.86
SentiwordNet	9.90	10.80	10.08	12.50

Table 3: ROUGE-2 results using one feature type.

Table 4 shows the results when combining the internal features with features from one external resource. We can see that the features from Word Embedding model outperform others, suggesting the effectiveness of this semantic similarity measure. Features from the LM scores and Wiki are also quite useful. Wiki pages are extracted for the test topic itself, therefore they provide topic relevant background information. The LM score features are extracted from large amounts of news article data, and are good representation of the general importance of bigrams for the test domain. In contrast, WordNet information is collected from a more general aspect, which may not be a very good choice for this task. Also notice that even though the features from DBpedia and sentiwordnet do not perform well by themselves, after the combination with internal features, there is significant improvement. This proves that the features from DBpedia and sentiwordnet provide additional information not captured by the internal features from the documents.

	2008	2009	2010	2011
Internal	10.40	11.76	10.42	12.91
+LM	10.76	12.03	10.80	13.11
+Word Embedding	10.92	12.12	10.85	13.24
+Wikipedia	10.81	12.08	10.76	13.17
+WordNet	10.68	11.96	10.71	12.99
+SentiwordNet	10.60	11.96	10.63	12.96
+DBpedia	10.69	12.00	10.70	13.07

Table 4: ROUGE-2 results using internal features combined with features from just one external resource.

Table 5 shows the results when adding features one by one. The order is based on its individual impact when combined with internal features. The results show that Wiki, LM and DBpedia features give more improvement than WordNet and SentiWordNet features. This shows the different impact of the external resources. We can see there is consistent improvement when more features are added.

	2008	2009	2010	2011
1: Internal +Word Embedding	10.92	12.12	10.85	13.24
2: 1+Wiki	11.22	12.25	11.15	13.47
3: 2+LM	11.41	12.41	11.37	13.68
4: 3+DBpedia	11.65	12.60	11.61	13.77
5: 4+WordNet	11.75	12.67	11.70	13.90
6: 5+SentiWordNet	11.84	12.77	11.78	13.97

Table 5: ROUGE-2 results using features incrementally combined.

Table 6 shows the feature ablation results, that is, each row means that the corresponding features are

excluded and the system uses all the other features. This set of experiments again shows that the external features like Word Embedding model based on large corpus and Wiki resource are very useful. Without using them, the system has the biggest performance degradation compared to the best result.

	2008	2009	2010	2011
-Internal	11.34	12.41	11.42	13.71
-Word Embedding	11.29	12.25	11.36	13.55
-Wiki	11.35	12.38	11.38	13.58
-LM	11.40	12.39	11.42	13.61
-DBpedia	11.50	12.47	11.47	13.71
-WordNet	11.67	12.64	11.64	13.80
-SentiWordNet	11.75	12.67	11.70	13.90

Table 6: ROUGE-2 results when leaving out each feature type.

5.2.4 Distribution of Correct Bigrams After Feature Weighting

In the next experiment we analyze the distribution of the correct bigrams from the ranked bigrams using different features in order to better evaluate their impact on bigram weighting. We rank all the bigrams in descending order according to the estimated weight, then calculate the number of correct bigrams (i.e., the bigrams in human generated summary) in Top10, 30, 50 and 80. The more correct bigrams appear on the top of the list, the better our features estimate the importance of the bigrams. We conducted this experiment using four systems: the system only with internal features, only with Word Embedding features, with combination of internal and Word Embedding features, and with all the features. Figure 2 shows the results of this experiment on TAC 2008 data. The pattern is similar on the other three years' data. The results show that systems with better ROUGE-2 value indeed can assign higher weights to correct bigrams, allowing the ILP decoding process to select these bigrams, which leads to a better sentence selection.

5.2.5 Joint Learning Results

Finally we evaluate the effectiveness of our proposed joint learning approach. For comparison, we implement a pipeline method, where we use the bigram's document frequency as the target value to train a regression model, and during testing use the model's predicted value as the weight in the ILP framework. Table 7 compares the results using the joint learning method and this pipeline approach. We only show the results using the system with all the features due to limited space. We can see that our joint method outperforms the pipeline system based on ROUGE-2 measurement, indicating that weights are better learned in the joint process that takes into account both bigram and sentence selection.

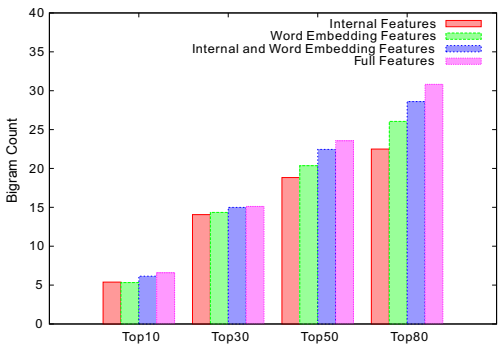

Figure 2: Distribution of correct bigrams in Top-n weighted bigrams from four systems.

System	2008	2009	2010	2011
Pipeline System	11.60	12.64	11.56	13.65
Joint Model	11.84	12.77	11.78	13.97

Table 7: ROUGE-2 results using different training strategies.

6 Conclusions

In this paper, we adopt the ILP based summarization framework, and propose methods to improve bigram concept selection and weighting. We use syntactic information to filter and select bigrams, various external resources to extract features, and a joint learning process for weight training. Our experiments in the TAC data sets demonstrate that our proposed methods outperform other state-of-the-art results. Through the analysis, we found the external resources are helpful to estimate the bigram importance and thus improve the summarization performance. While in summarization research, optimization-based methods have already rivaled other approaches in performance, the task is

far from being solved. Our analysis revealed that there are at least three points worth mentioning. First, using external resources contributes to the improved performance of our method compared to others that only use internal features. Second, employing and designing sophisticated features, especially those that encode background knowledge or semantic relationship like the word embedding features from a large corpus we used, will enable language concepts to be distinguished more easily in the presence of a large number of candidates. Third, one limitation of the use of the external resources is that they are not always available, such as the pairwise news articles along with the human generated summaries, and the relevant Wiki pages. While much recent work has focused on algorithmic development, the summarization task needs to have a deeper "understanding" of a document in order to reach the next level of performance. Such an understanding can be facilitated by the incorporation of background knowledge, which can lead to significant summarization performance improvement, as demonstrated in this study.

Acknowledgments

We thank the anonymous reviewers for their detailed and insightful comments on earlier drafts of this paper. The work is partially supported by NSF award IIS-0845484 and DARPA Contract No. FA8750-13-2-0041. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of LREC*.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*.

Sashka T. Davis, John M. Conroy, and Judith D. Schlesinger. 2012. Occams - an optimal combinatorial covering algorithm for multi-document summarization. In *Proceedings of ICDM*.

Dan Gillick, Benoit Favre, and Dilek Hakkani-Tur. 2008. The ICSI summarization system at tac 2008. In *Proceedings of TAC*.

Dan Gillick, Benoit Favre, Dilek Hakkani-Tur, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/UTD summarization system at tac 2009. In *Proceedings of TAC*.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of ACL*.

Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of EACL*.

Houping Jia, Xiaojiang Huang, Tengfei Ma, Xiaojun Wan, and Jianguo Xiao. 2010. Pkutm participation at tac 2010 rte and summarization track. In *Proceedings of TAC*.

Jay J Jiang and David W Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

Samir Khuller, Anna Moss, and Joseph Seffi Naor. 1999. The budgeted maximum coverage problem. *Information Processing Letters*.

Judith L. Klavans and Min-Yen Kan. 1998. Role of verbs in document analysis. In *Proceedings of the ACL*.

Niraj Kumar, Kannan Srinathan, and Vasudeva Varma. 2010. An effective approach for aesop and guided summarization task. In *Proceedings of TAC*.

Chen Li, Fei Liu, Fuliang Weng, and Yang Liu. 2013a. Document summarization via guided sentence compression. In *Proceedings of the EMNLP*.

Chen Li, Xian Qian, and Yang Liu. 2013b. Using supervised bigram-based ilp for extractive summarization. In *Proceedings of ACL*.

Chen Li, Yang Liu, Fei Liu, Lin Zhao, and Fuliang Weng. 2014. Improving multi-documents summarization by sentence compression based on expanded constituent parse trees. In *Proceedings of EMNLP*.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Proceedings of NAACL*.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of ICML*.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Proceedings of ACL*.

Patrice Lopez and Laurent Romary. 2010. Humb: Automatic key term extraction from scientific articles in grobid. In *Proceedings of the international workshop on semantic evaluation*.

Andre F. T. Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the ACL Workshop on Integer Linear Programming for Natural Language Processing*.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of ECIR*.

Olena Medelyan, Eibe Frank, and Ian H Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the EMNLP*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*.

Jun-Ping Ng, Praveen Bysani, Ziheng Lin, Min-Yen Kan, and Chew-Lim Tan. 2012. Exploiting category-specific information for multi-document summarization. In *Proceedings of COLING*.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING-ACL*.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

Hiroya Takamura and Manabu Okumura. 2009. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of EACL*.

Zhongyu Wei and Wei Gao. 2014. Utilizing microblogs for automatic news highlights extraction. In *Proceedings of COLING*.

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *Proceedings of EMNLP-CoNLL*.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of ACL*.

Wen-Tau Yih, Joshua Goodman, and Vitor R Carvalho. 2006. Finding advertising keywords on web pages. In *Proceedings of international conference on World Wide Web*.