PPSGen: Learning-Based Presentation Slides Generation for Academic Papers

Yue Hu and Xiaojun Wan

Abstract—In this paper, we investigate a very challenging task of automatically generating presentation slides for academic papers. The generated presentation slides can be used as drafts to help the presenters prepare their formal slides in a quicker way. A novel system called PPSGen is proposed to address this task. It first employs the regression method to learn the importance scores of the sentences in an academic paper, and then exploits the integer linear programming (ILP) method to generate well-structured slides by selecting and aligning key phrases and sentences. Evaluation results on a test set of 200 pairs of papers and slides collected on the web demonstrate that our proposed PPSGen system can generate slides with better quality. A user study is also illustrated to show that PPSGen has a few evident advantages over baseline methods.

Index Terms—Abstracting methods, text mining

1 Introduction

PRESENTATION slides have been a popular and effective means to present and transfer information, especially in academic conferences. The researchers always make use of slides to present their work in a pictorial way on the conferences. There are many softwares such as Microsoft PowerPoint and OpenOffice to help researchers prepare their slides. However, these tools only help them in the formatting of the slides, but not in the content. It still takes presenters much time to write the slides from scratch. In this work, we propose a method of automatically generating presentation slides for academic papers. We aim to automatically generate well-structured slides and provide such draft slides as a basis to reduce the presenters' time and effort when preparing their final presentation slides.

Academic papers always have a similar structure. They generally contain several sections like abstract, introduction, related work, proposed method, experiments and conclusions. Although presentation slides can be written in various ways by different presenters, a presenter, especially a beginner, always aligns slides sequentially with the paper sections when preparing the slides. Each section is aligned to one or more slides and one slide usually has a title and several sentences. These sentences may be included in some bullet points. Our method attempts to generate draft slides of the typical type mentioned above and helps people to prepare their final slides.

Automatic slides generation for academic papers is a very challenging task. Current methods generally extract objects like sentences from the paper to construct the slides. In contrast to the short summary extracted by a summarization system, the slides are required to be much more structured and much longer. Slides can be divided into an ordered sequence of parts. Each part addresses a specific topic and these topics are also relevant to each other. Generally speaking, automatic slide generation is much more difficult than summarization. Slides usually not only have text elements but also graph elements such as figures and tables. But our work focuses on the text elements only.

In this study, we propose a novel system called PPSGen to generate well-structured presentation slides for academic papers. In our system, the importance of each sentence in a paper is learned by using the support vector regression (SVR) model with a number of useful features, and then the presentation slides for the paper are generated by using the integer linear programming (ILP) model with elaborately designed objective function and constraints to select and align key phrases and sentences.

Experiments on a test set of 200 paper-slides pairs indicate our method can generate slides with better quality than the baseline methods. Using the ROUGE toolkit and the pyramid evaluation, the slides generated by our method can get better ROUGE scores and pyramid scores. Moreover, based on a user study, our slides can get higher rating scores by human judges in both content and structure aspects. Therefore, our slides are considered a better basis for preparing the final slides.

The rest of this paper is organized as follows. Related work is introduced in Section 2. We describe our method in detail in Section 3. We show the experiment results in Section 4 and conclude our work in Section 5.

2 Related Work

2.1 Slides Generation

Automatic slides generation for academic papers remains far under-investigated nowadays. Few studies directly research on the topic of automatic slides generation. Utiyama and

- *The authors are with the Institute of Computer Science and Technology, The MOE Key Laboratory of Computational Linguistics, Peking University, Beijing 100871, China.*
  E-mail: {ayue.hu, wanxiaojun}@pku.edu.cn.

Manuscript received 17 June 2013; revised 31 Aug. 2014; accepted 8 Sept. 2014. Date of publication 21 Sept. 2014; date of current version 3 Mar. 2015.
Recommended for acceptance by P. G. Ipeirotis.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2014.2359652

Hasida [1] attempted to automatically generate slides from input documents annotated with the GDA tagset.[1] GDA tagging can be used to encode semantic structure. The semantic relations include grammatical relations such as subject, thematic relations such as agent, patient, and rhetorical relations such as cause and elaboration. They first detect topics in the input documents and then extract important sentences relevant to the topics to generate slides.

Yasumura et al. [2] introduced a support system for making slides from technical papers. The inputs of the system are academic papers in LATEX format. The system calculates the weights of the terms in the paper using TF*IDF scores. Using the term weights, objects in the paper like sentences, tables etc. are also weighted. Based on the weights of the objects, the system decides the number of the objects like sentences to be extracted for each section in the paper and then generate the slides using a slide composition template which can be edited by the users.

Shibata and Kurohashi [3] proposed a method to automatically generate slides from raw texts. Clauses and sentences are considered as discourse units and coherence relations between the units such as list, contrast, topic-chaining and cause are identified. Some of clauses are detected as topic parts and others are regarded as non-topic parts. These different parts are used to generate the final slides based on the detected discourse structure and some heuristic rules.

Hayama et al. [4], Kan [5] and Beamer and Girju [6] studied the problem of aligning technical papers and presentation slides. Hayama et al. used a variation of the Hidden Markov Model (HMM) to align the text in the slides to the most likely section in the paper, which also used the additional information of titles and position gaps. Kan [5] applied a modified maximum similarity method to do the monotonic alignments and trained a classifier to detect slides which should not be aligned. Beamer and Girju [6] compared and evaluated four different alignment methods that were combined by methods such as TF-IDF term weighting and query expansion.

Masum et al. [7], [8] proposed a system named automatic report to presentation (ARP) which constructs a topic-specific report and a presentation on a topic or search phrases given by a user. The system retrieves webpages relevant to the disambiguated query using multiple search engines. Headings and text chunks are extracted from webpages and used to build the report. A presentation is generated by randomly selecting up to five lines from each head-text tuple, two lines from the top, one in the middle and the other two lines from the end of the text chunk.

Sravanthi et al. [9] investigated automatic generation of presentation slides from technical papers in LATEX. A query specific extractive summarizer QueSTS is used to extract sentences from the text in the paper to generate slides. QueSTS transfers the input text to an integrated graph (IG) where a sentence represents a node and edges exist between the nodes that the sentences corresponding to them are similar. The weights of the edges are calculated as cosine similarity between the sentences. More details can be found in [10].

Different from the above approaches which simply select and place objects like sentences on the slides, we attempt to generate more structured slides based on learning strategies. The slides contain not only sentences but also key phrases aligned to the sentences. Key phrases are used as the bullet points and sentences relevant to the phrases are placed below them.

2.2 Scientific Article Summarization

The goal of scientific article summarization is to generate a short summary for a given scientific article or article set. Early works including [11], [12], [13] tried to use various features specific to scientific text (e.g., rhetorical clues features).

Citation information has already shown its effectiveness for summarization of the scientific articles. Various works including [14], [15], [16], [17], [18], [19], [20], [21] employed citation information for the scientific article summarization. Earlier work [22] indicated that citation sentences may contain important concepts that can give useful descriptions of a paper.

Agarwal et al. [23] introduced an unsupervised approach to the problem of multi-document scientific article summarization. The input is a list of papers cited together within the same source article. The key point of this approach is a topic based clustering of fragments extracted from each co-cited article.

Yeloglu et al. [24] compared four different approaches for multi-document scientific articles summarization: MEAD, MEAD with corpus specific vocabulary, LexRank and W3SS.

2.3 Document Summarization

The task of document summarization aims to generate a very short summary for a given document or document set. Various methods have been proposed for document summarization, including rule-based methods [25], [26], graph-based methods [27], [28], [29], learning-based methods [30], [31], [32], [33], ILP-based methods [34], [35], [36], [37], [38], [39], etc.

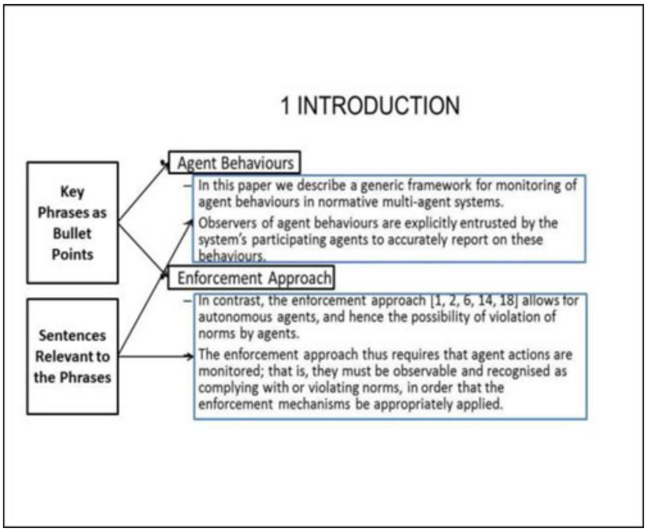
Recently support vector regression and ILP have been used widely in the task of summarization. Ouyang et al. [32] and Galanis and Malakasiotis [33] used SVR to train and learn the sentence importance score. McDonald [34] proposed the first ILP method for summarization. It constructed summaries by maximizing the importance of the selected sentences and minimizing their pairwise similarity.

Gillick et al. [35], [36] and Berg-Kirkpatrick et al. [37] introduced and adopted an ILP method based on the notion of 'concepts' which are actually bigrams. Each concept (bigram) has a weight w. Each sentence is considered to consist of a set of concepts and the ILP approach aims to maximize the weights of the concepts covered by a selection of sentences.

Woodsend and Lapata [38] also adopted methods based on ILP to extract summary. The object function of the ILP model combines the importance of the bigrams in the summary's sentences, the salience of the parse tree nodes of the summary's sentences and a unigram language model which penalizes sentences containing words that are likely to appear in summaries.

Galanis et al. [39] used a method based on both SVR and ILP to deal with multi-document summarization which is most relevant to our work. The paper used an SVR model

[1] http://www.i-content.org/GDA/tagset.html



1 INTRODUCTION

Agent Behaviours

- In this paper we describe a generic framework for monitoring of agent behaviours in normative multi-agent systems.
  Observers of agent behaviours are explicitly entrusted by the system's participating agents to accurately report on these behaviours.

Enforcement Approach

- In contrast, the enforcement approach [1, 2, 6, 14, 18] allows for autonomous agents, and hence the possibility of violation of norms by agents.
  The enforcement approach thus requires that agent actions are monitored; that is, they must be observable and recognised as complying with or violating norms, in order that the enforcement mechanisms be appropriately applied.

Key Phrases as Bullet Points
Sentences Relevant to the Phrases

Fig. 1. A sample slide we generate.

to predict the importance of the sentences in the documents and applied two similar versions of the ILP model to select the sentences to generate the multi-document summary.

The above approaches all deal with the tasks of traditional summarization and scientific article summarization. However, slides generation is much different from traditional summarization and scientific summarization. They simply select several sentences from the documents, while slides generation is much more complicated. Our proposed approach not only selects a number of important sentences but also the phrases corresponding to the sentences. After the selection of sentences and phrases, we can construct well-structured slides.

3 Problem Definition and Corpus

3.1 Problem Definition

In our work, we aim to automatically generate presentation slides for academic papers. We need to generate well-structured slides as the draft slides for a presenter to prepare the final slides.

There are various kinds of slides which are made by Microsoft PowerPoint and OpenOffice. They can be much different in styles and we obviously cannot consider all kinds of styles. So before introducing our method, we need to address the style of slides we generate.

A beginner usually prepares slides which are sequentially aligned with the paper. One section in the paper is generally aligned to one or more slides. One slide usually includes several bullet points and sentences that explain the corresponding bullet points. It is reasonable to use that style of slides that beginners always use to make draft slides and we regard it well-structured because it uses pairs of bullet points and sentences to address important points and makes it easy for the reader to handle the points. From Fig. 1, we can have a glance at the style of the slides we generate. Here, key phrases "Agent Behaviors" and "Enforcement Approach" are set as the bullet points. The sentences relevant to the key phrases are placed below the corresponding bullet points.

In this work, we only consider the text elements in the paper. Other elements such as tables and figures are not included in the generated slides. Though tables and figures are useful in the slides, we ignore them to simplify the problem and better focus on the generation of the text elements.

3.2 Corpus and Preprocessing

To learn how humans generate slides from academic papers, we build a corpus that contains pairs of academic papers and their corresponding slides. Many researchers in the computer science field place their papers and the corresponding slides together in their homepages. The homepages' URLs are obtained by crawling Arnetminer.[2] After downloading the homepages, we use several strict patterns to extract the links of the papers and the associated slides and download the files to build the dataset. We collect more than 2,000 pairs. After cleaning up the incorrect pairs, we have 1,200 paper-slides pairs.

The papers are all in PDF format and the slides are in either PDF or PowerPoint format. For the papers, we extract their texts by using PDFlib[3] and detect their physical structures of paragraphs, sections and sections by using ParsCit.[4] A custom XML format is used to describe this structure. For the slides, we also extract their texts and physical structures like sentences, titles, bullet points, etc. We use xpdf[5] and the API provided by Microsoft Office to deal with the slides in PDF and PowerPoint formats, respectively. The slides are transformed to a predefined XML format as well.

4 Our Proposed Method

4.1 Overview

In this paper, we propose a system to automatically generate slides that have good structure and content quality from academic papers. The architecture of our system is shown in Fig. 2. We use the SVR-based sentence scoring model to assign an importance score for each sentence in the given paper, where the SVR model is trained on a corpus collected on the web. Then, we generate slides from the given paper by using ILP. More details of each part will be discussed in the following sections.

4.2 Sentence Importance Assessment

In our proposed PPSGen system, sentence importance assessment is one of the two key steps, which aims to assign an importance score to each sentence in the given paper. The score of each sentence will be used in the slides generation process. In this study, we introduce a few useful features and propose to use the support vector regression model to achieve this goal.

4.2.1 Support Vector Regression Model

Here we briefly introduce the SVR model [40]. Let $\{u_i, y_i\}_{i=1}^{N}$ $(u_i \in R^d, y_i \in R)$ be a set of data points. The support vector regression model aims to learn a function $f(u)$ which has the following form:

[2] http://arnetminer.org
[3] http://www.pdflib.com/
[4] http://aye.comp.nus.edu.sg/parsCit/
[5] http://www.foolabs.com/xpdf/

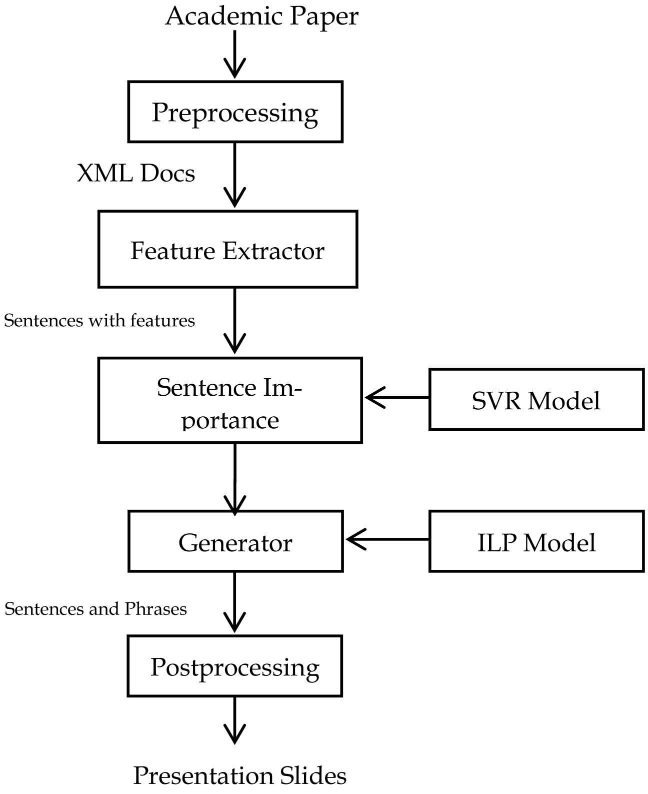


Fig. 2. System architecture.

$$f(u) = w * \varphi(u) + b, \tag{1}$$

where φ(u) represents the high-dimensional feature spaces which are nonlinearly transformed from u. It chooses the optimum function f(u) to minimizing the structure risk function below:

$$\frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N} L(y_i, f(u_i)). \tag{2}$$

The first term is the regularization term to avoid overfitting. The second term is the empirical error measured by ε-insensitivity loss function which is defined as $\mathrm{L(x)} = |\mathrm{x}| - \varepsilon$ if $|\mathrm{x}| > \varepsilon$, and $\mathrm{L(x)} = 0$, otherwise. The regularization constant C and the radius ε can be set by the user.

After introducing the kernel function $k(u_i, u_j)$ and solving the optimization problem mentioned above, we can get

$$f(u) = \sum_{i=1}^{N} \beta_i k(u_i, u) + b. \tag{3}$$

In our case, *u* is the feature vector of the sentence and *y* is the importance score of the sentence. We use LIBSVM [41] with the RBF kernel to implement the SVR model.

We need to predict the importance score of each sentence for sentence selection in slides generation. The reason why we use the SVR model instead of the classification model is that the regression score is finer to be used for sentence selection than the coarse binary category.

4.2.2 Training Data Construction and Model Learning

To construct training data based on the paper-slides pairs, we apply a similarity scoring method to assign the importance scores to the sentences in a paper. The main hypothesis is that the sentences in the slides should represent the substance of the corresponding paper. The sentences in the paper which are more similar to the sentences in the slides should be considered more important and higher scores should be assigned to them using the scoring method.

Thus, we define the sentence's importance score in the paper as follows:

$$score(s) = \max_{s_i^* \in S^*}(sim(s, s_i^*)), \tag{4}$$

where *s* is a sentence in the paper, S^* is the set of the sentences in the corresponding slides and s_i^* is a sentence in S^*. The standard cosine measure is used as the similarity function $sim(s, s_i^*)$. So the sentence's importance score is set as the maximum similarity between the sentence and any sentence in the corresponding slides.

Intuitively, a sentence with a higher maximum similarity is closer to one sentence in the author-written slides. Since the author-written slides contain the sentences that human authors considered most important, a sentence with a higher score is most likely to be important, too.

We adopt the maximum similarity instead of the overall similarity with all the sentences in the slides or the average similarity with each sentence in the slides. The motivation is that slides can be generally divided into several parts and each part may be relevant to one section in the paper. The sentences in a specific section should be more similar to the corresponding part in the slides and less similar to the other parts. Therefore, it is more reasonable to use the maximum similarity to assign the importance scores of the sentences.

Each sentence in a paper is represented by a set of features. In this study, we make use of the following features for each sentence *s*:

1) *Similarity with the titles*. We consider three types of titles: paper title, section titles and section titles. Only the titles of the section and section which contain the sentence are used. We use the cosine similarity values between the sentence and different types of titles as different features. Stop words are removed and all the words are stemmed in the similarity calculation. Intuitively the sentences that have higher similarity with the titles should be more likely to be selected.
2) *Word overlap with the titles*. It is the number of words shared by the sentence and the set of words of all titles, including all three types of titles mentioned above.
3) *Sentence position,* which is computed as follows.

$$SP(s) = \frac{pos(s, sec(s))}{|sec(s)|}, \tag{5}$$

where $pos(s, sec(s))$ is the position of sentence *s* in its section $sec(s)$, $|sec(s)|$ is the number of sentences in $sec(s)$. In general, the first and last few sentences are always more important.
4) *Sentence's parse tree information*. The features are extracted from the sentence's parse tree. It includes

the number of noun phrases and verb phrases, the number of sub-sentences and the depth of the parse tree. We use the OpenNLP library[6] for sentence parsing.

5) *Stop words percentage*. It is the percentage of the stop words in the total word set of the sentence *s*. Intuitively the sentences that have high percentage of the stop words are less likely to be important.
6) Other features including the length of sentence *s*, the number of words after removing stop words and the average length of sentences of the section, section or paragraph that contains the sentence.

All the features mentioned above are scaled into [–1, 1]. Based on the features and importance scores of the sentences in the training data, we can learn an SVR model, and then apply the model to predict an importance score for each sentence in any paper in the test set. The score indicates the possibility of a sentence to be selected for making slides.

4.3 Slides Generation

After getting the predicted importance score for each sentence in the given paper, we exploit the integer linear programming method to generate well-structured slides by selecting and aligning key phrases and sentences.

Unlike those methods [1], [2], [9] that generate slides by simply selecting important sentences and placing sentences on the slides, we select both key phrases and sentences to construct well-structured slides. We use key phrases as the bullet points and sentences relevant to the phrases are placed below the bullet points.

In order to extract the key phrases, chunking implemented by the OpenNLP library is applied to the sentences and noun phrases are extracted as the candidate key phrases.

We define two kinds of phrases: global phrases and local phrases. Any unique phrase in an article is a global phrase, and a local phrase means a global phrase in a particular section. For example, "SVR" is a global phrase of this paper, while its appearances in different sections are considered different local phrases. "SVR {Introduction}" and "SVR {Our Proposed Method}" denote different local phrases, and they represent the appearances of "SVR" in Sections 1 and 4 of this paper, respectively. So a global phrase that appears in different sections can correspond to a few local phrases. Since an important phrase is always used in many different sections, a global phrase that corresponds to more local phrases should be regarded to be more important and more likely to be selected. Thus, we use the local phrases to generate the bullet points directly for different sections and use the global phrases to address the importance differences between different unique phrases. All the phrases are stemmed and stop words are removed. Moreover, the noun phrases that appear only once in the paper are discarded.

The object function and constraints of our proposed ILP model are presented as follows:

$$\max_{lp,x} \lambda_1 \sum_{i=1}^{n} \frac{l_i}{L_{max}} w_i x_i + \lambda_2 \sum_{i=1}^{|B|} \frac{c_{b_i} b_i}{|B^*|} + \lambda_3 \sum_{i=1}^{n} \frac{w_i}{n} y_i, \quad (6)$$

subject to:

$$\sum_{i=1}^{n} l_i x_i \leq L_{max}, \quad (7)$$

$$\sum_{lp_j \in LP_i} lp_j \geq x_i, for\ i = 1, \ldots n, \quad (8)$$

$$\sum_{s_i \in S_j} x_i \geq lp_j, for\ j = 1, \ldots |LP|, \quad (9)$$

$$\sum_{lp_j \in LP_k} lp_j \geq y_k, for\ k = 1, \ldots n, \quad (10)$$

$$\sum_{b_m \in B_i} b_m \geq |B_i| x_i, for\ i = 1, \ldots n, \quad (11)$$

$$\sum_{s_i \in S_m} x_i \geq b_m, for\ m = 1, \ldots |B|, \quad (12)$$

$$\sum_{lp_j \in GP_t} lp_j \geq gp_t, for\ t = 1, \ldots |GP|, \quad (13)$$

$$gp_t \geq lp_j, for\ \forall lp_j \in GP_t; t = 1, \ldots |GP| \quad (14)$$

$$\sum_{i=1}^{|GP|} gp_i * 2 \leq \sum_{i=1}^{n} x_i, \quad (15)$$

$$x_i, lp_j, y_k, b_m, gp_t \in \{0, 1\}, \forall i, j, k, m, \quad (16)$$

where:

w_i – the importance weight of sentence s_i which is computed by using the SVR model;
n – the number of sentences
l_i – the length of sentence s_i
x_i – the variable that indicates whether sentence s_i is included in the slides
lp_j – the variable that indicates whether local phrase lp_j is included in the slides
gp_t – the variable that indicates whether global phrase gp_t is included in the slides
GP_t – the set of local phrases relevant to global phrase gp_t
y_k – the variable that indicates whether sentence s_k contains at least one selected local phrase
b_m – the variable that indicates whether bigram b_m is included in the slides
L_{max} – the maximum length of the slides
c_{b_i} – the count of the occurrences of bigram b_i in the paper
B^* – the total set of bigrams in the paper
B – the set of unique bigrams after removing duplicated bigrams
LP – the set of the total local phrases
GP – the set of the total global phrases

[6] http://opennlp.apache.org/

The object function contains three parts. The explanation of each part is below:

1) The first part maximizes the overall importance score of the generated slides. It sums the importance scores of the selected sentences. Rather than simply calculating the sum of the scores, we add the sentence length as a multiplication factor in order to penalize the very short sentences. In addition, adding the sentence length as the multiplication factor won't lead the model to select the very long sentences. The total length of the sentences selected is fixed. So if the model tends to select the longer sentences, the less sentences can be selected. The model needs to make a tradeoff between the number and the average length of the sentences selected. This issue is addressed in [39]. Galanis et al. [39] show the version of the ILP model that does not consider the length of the sentences gets worse performance. So we consider the length of the sentences in our model. We use this part to estimate the overall importance of the generated slides.
2) The second part maximizes the total counts of the bigrams in the paper which also appear in the slides. Bigrams that appear more times should be considered more important and more likely to be selected. When more unique bigrams are presented in the slides, the sentences in the slides are less redundant and the slides can be more diverse. And when more important bigrams are included in the slides, the sentences selected are more important and slides can get better quality. We use this part to maximize the diversity of the slides.
3) The last part aims to maximize the weighted coverage of the key phrases selected and determine which phrases should be selected. We say a sentence is covered by a phrase when this sentence contains the phrase. High-quality slides should cover the content in the paper as much as possible. According to our defined type of slides which include phrases and sentences below the phrases, we should select the phrases that are relevant to more sentences. We consider not only the count of the sentences covered by the selected phrases but also the overall importance score of the covered sentences. We have introduced two kinds of phrases: global phrases and local phrases above. Here we use the global phrase to count the sentences covered. When one phrase is selected, the sentences that contain this phrase in different sections are counted.

All the terms in the object function are normalized to [0, 1] by using the maximum length L_{max}, the total number of bigrams |B| and the number of the sentences n, respectively. The values of λ_1, λ_2 and λ_3 are parameters for tuning the three parts and we set $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

The explanation of each constraint is below:

Constraint (7). It guarantees that the total word count of the slides does not exceed L_{max}.

Constraint (8). LP_i is the set of local phrases that sentence s_i contains. If a sentence s_i is selected ($x_i = 1$), then at least one of its local phrases must also be selected, i.e., $\sum_{lp_j \in LP_i} lp_j \geq 1$. If sentence s_i is not selected ($x_i = 0$), some of its local phrases may still be selected, i.e., $\sum_{lp_j \in LP_i} lp_j \geq 0$. It ensures that when sentence s_i is selected, at least one local phrase in LP_i is selected.

Constraint (9). S_j is the set of sentences that contains local phrase lp_j. If a local phrase $\mathrm{lp_j}$ is selected ($lp_j = 1$), then at least one sentence in S_j must be selected to generate the slides. If local phrase lp_j is not selected ($lp_j = 0$), then sentences that contain lp_j may be selected. It ensures that when local phrase lp_j is selected, at least one sentence in S_j is selected.

Constraint (10). LP_k is the set of local phrases that sentence $s_k$contains. If at least one local phrase in LP_k is selected ($\sum_{lp_j \in LP_k} lp_j \geq 1$), the value of y_k must be set to 1 because the ILP solution should maximize the object function and must set y_k to 1 if it can. If no local phrase in LP_k is selected ($\sum_{lp_j \in LP_k} lp_j = 0$), y_k must be set to 0. It ensures that y_k is set to 1 only when at least one local phrase in LP_k is selected.

Constraints (11), (12). These two constraints are similar to constraints (8) and (9), respectively. B_i is the set of bigrams that sentence s_i contains. $|B_i|$ is the count of the set B_i. S_m is the set of sentences that include bigram b_m. When constraint (11) holds, all the bigrams that s_i has can be ensured to be selected if s_i is selected and some of the bigrams that s_i has may be selected if s_i is not selected. Constraint (12) guarantees at least one sentence in S_m is selected, i.e., $\sum_{s_i \in S_m} x_i \geq 1$ if b_m is selected ($b_m = 1$). If b_m is not selected, sentences in S_m may still be selected and $\sum_{s_i \in S_m} x_i \geq 0$. These two constraints are used to prevent that not all bigrams in one sentence selected are counted or one bigram selected cannot find one sentence selected that contains it.

Constraint (13). The definitions of *gp* and *lp* are mentioned before. If a global phrase gp_t is included in the slides ($gp_t = 1$), then at least one local phrase relevant to gp_t must be selected, i.e., $\sum_{lp_j \in GP_t} lp_j \geq 1$. If global phrase gp_t is not selected, then no local phrase relevant to gp_t should be selected, i.e., $\sum_{lp_j \in GP_t} lp_j = 0$. This constraint guarantees that if a global phrase is selected, at least one corresponding local phrase is selected.

Constraint (14). If a local phrase lp_j is selected ($lp_\mathrm{j} = 1$), then the corresponding global phrase gp_t must be selected, i.e., $gp_t = 1$. If a local phrase lp_j is not selected ($lp_j = 0$), the corresponding global phrase gp_t may still be selected ($gp_t \geq 0$), for other local phrases relevant to gp_t may be selected. In the two cases, the constraint holds. This constraint ensures that if a local phrase is selected, the corresponding global phrase is also selected.

Constraint (15). It guarantees the total number of the selected global phrases is less than half the number of the sentences selected. Using this constraint, we can avoid extracting too many key phrases. In our experiment, we find out the factor two is a reasonable value. Here we use the count of global phrases instead of local phrases. The motivation is that phrases appear in several sections are considered more important. The ILP solution tends to select these important phrases to maximize the object function. We reward these important phrases in this way.

In addition, we develop two simplified ILP models, called SILP1 and SILP2 to compare with the above ILP model (abbr. ILP). The object functions of SILP1 and SILP2 are shown below:

$$\max_{lp,x} \lambda_1 \sum_{i=1}^{n} \frac{l_i}{L_{max}} w_i x_i + \lambda_2 \sum_{i=1}^{|B|} \frac{b_i}{|B|} + \lambda_3 \sum_{i=1}^{n} \frac{w_i}{n} y_i \tag{17}$$

$$\max_{lp,x} \lambda_1 \sum_{i=1}^{n} \frac{l_i}{L_{max}} w_i x_i + \lambda_2 \sum_{i=1}^{|B|} \frac{c_{b_i} b_i}{|B^*|} + \lambda_3 \sum_{i=1}^{n} \frac{1}{n} y_i. \tag{18}$$

We change the second and third part of the original object function in the two simplified models, respectively. The constraints do not need to be changed. In the first simplified ILP model we only count the number of bigrams that appear in the slides and ignore the bigrams' weights. In the second simplified ILP model we count the number of the sentences that are covered by the selected phrases and ignore the sentences' score.

The ILP model is applied to the whole paper once and the method does not assign the number of slides for each section explicitly. It is hard to decide the length of slides for each section. It is inappropriate to set the length of slides for each section just according to the section's length, for slides generally are not organized in this way and the length of slides may not correspond with the section's length. That is the reason why we apply the ILP model to the whole paper once instead of each section or subsection. We just leave this work to the ILP model that we design. By maximizing the object function, the ILP model can assign the number of slides well. We think it is a better way to organize the generated slides.

Using the ILP model, we can obtain the aligned key phrases and sentences to be included in the slides. The titles of slides are set by using the titles of the corresponding sections. When we generate the slides, we regard a phrase and its corresponding sentences as a group. The length of the group is defined as the total length of the sentences. We decide the number of the groups placed on one slide according to the length of the groups. We place at most two groups on one slide. We solve the above optimization problem by using the IBM CPLEX optimizer.[7] It generally takes about 10 seconds to solve the problem. Then the draft slides are generated by using the API provided by Microsoft Office.

5 Evaluation

5.1 Evaluation Setup

In order to set up our experiments, we divide our dataset which contains 1,200 pairs of paper and slides into three parts: 800 pairs for training, 200 pairs for test and the other 200 pairs for parameter tuning. The SVR regression model with the RBF kernel in LIBSVM is trained on the training data and applied to the test data. Then the ILP model is used to generate the slides. The maximum word count of the slides is set to 15 percentage of the total word count of the paper. In our experiment, 15 percentage is a reasonable value because the generated slides can have an appropriate length. Then the parameter values of the ILP methods are tuned on the parameter tuning data. The values of parameters λ_1, λ_2 and λ_3 in our methods are set to 0.3, 0.4 and 0.3, respectively.

We implement four baseline methods for comparison with our proposed method:

1) *TF-IDF based method*. This method is used by [2] based upon the TF-IDF scores, which extracts sentences for each section or section. The IDF scores are calculated on the corpus we collect. The sentences that have larger TF-IDF scores are selected to generate the slides.
2) *MEAD based method*. MEAD[8] [42] is the most elaborate and publicly available platform for multi-lingual summarization. It implements multiple summarization methods including position-based, centroid-based, length-based and query-based. A combination of these methods is adopted to extract the slides. MEAD is applied to the whole paper instead of each section or section.
3) *Random Walk based method*. In the Random Walk [43] method, sentences are regarded as nodes, the cosine similarities between sentences are assigned to be the weights of edges and the random walk method is employed to assign scores to the sentences. Here we apply the random walk method to each section, i.e., the sentences in one section and their relationship make up the graph that the random walk method applies to. The sentences that have larger scores are selected to generate the slides.
4) *C-lexrank***.** C-Lexrank [17] is a clustering-based model in which the cosine similarities of sentences pairs are used to build a network of sentences. Based on the similarity network, C-Lexrank employs [44], a hierarchical agglomeration algorithm which works by greedily optimizing the modularity for sparse graphs. At last, it calculates Lexrank within each cluster to find the most salient sentences, and select the sentences with respect to their salience from different clusters. Lexrank [45] first builds a lexical network, in which nodes are sentences and the lexical similarities are set as the weights of edges. Then the random walk is applied to the network like our random walk based baseline. C-Lexrank is applied to the whole paper.

All the slides extracted by the above baseline methods have the same lengths as those generated by our method. In the TF-IDF based and Random Walk baseline methods, we extract sentences for each section. So we need to decide the total length of the sentences selected for each section. For our method, the MEAD method and the C-Lexrank method, we do not need to do that because we apply the methods to the whole paper. The total length of sentences extracted for each section is determined by multiplying the maximum length by the ratio of the section's length to the paper's length.

In addition, we implement four different methods to compute the importance scores of sentences to compare with the SVR-based method with the maximum similarity

[7] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[8] http://www.summarization.com/mead/

in our method (abbr. Max-SVR). The first two are also based on SVR while the other two are not. They are described below:

1) *Overall-SVR*. This method use the overall similarity between the sentence and all the sentences in the slides as the scoring method which is shown as follows:

$$score(s) = sim(s, S^*), \tag{19}$$

where s is a sentence in the paper, S^* is the set of the sentences in the corresponding slides. The standard cosine measure is used as the similarity function $sim(s, S^*)$. SVR is then applied to learn and predict the importance scores of the sentences.

2) *Avg-SVR*. This method use the average similarity between the sentence and each sentence in the slides which is shown below:

$$score(s) = \underset{s_i^* \in S^*}{avg}\left(sim\left(s, s_i^*\right)\right). \tag{20}$$

The definitions of the symbols are also the same as above. Instead of using the overall similarity, we use the average similarity here.

3) *TF-IDF*. The TF-IDF scores are used as the importance scores of the sentences.

4) *Random Walk*. The Random Walk scores are used as the importance scores of the sentences. When calculating the Random Walk scores, we regard the sentences in the whole paper and their relationship as a graph, which is different from the Random Walk baseline mentioned above.

The scores are then normalized to [0, 1] as below to better fit the ILP model:

$$score^* = (score - score_{min})/(score_{max} - score_{min}) \tag{21}$$

After getting the scores, we apply the same ILP model to generate the draft slides.

Moreover, we train two different models to learn the importance scores of sentences. Both models use the maximum similarity as the scoring method like our SVR model and their features are the same as our SVR model. They are described below:

1) *SVM*. We treat the sentences whose similarities are above the threshold 0.5 as the positive training examples and the other sentences are regarded as the negative examples. The SVM classification model with the RBF kernel in LIBSVM is trained and then applied to the test data.

2) *Linear regression*. The linear regression model is trained on the training data and used to predict the importance scores of sentences in the test data.

In the above section, we mention two simplified ILP model: SILP1 and SILP2. We also compare their performance with our adopted ILP model in our experiments.

We use the ROUGE toolkit to evaluate the content quality of the generated slides. ROUGE [46] is a state-of-the-art automatic evaluation method based upon n-gram comparison, which is widely used in summarization evaluation. We use the F-Measure scores of ROUGE-1, ROUGE-2, ROUGE-SU4. ROUGE compares the generated text with the reference text. Here, the set of sentences in the author-written slides of the paper is regarded as the reference text. ROUGE-N is an n-gram based measure between a candidate text and a reference text. The recall oriented score, the precision oriented score and the F-measure score for ROUGE-N are computed as follows:

$$\begin{aligned} &ROUGE - N_{Recall} \\ &\quad = \sum_{S \in \{ReferenceText\}} \sum_{gram_n} Count_{match}(gram_n)/ \\ &\quad = \sum_{S \in \{ReferenceText\}} \sum_{gram_n} Count(gram_n) \end{aligned} \tag{22}$$

$$\begin{aligned} &ROUGE - N_{Precision} \\ &\quad = \sum_{S \in \{ReferenceText\}} \sum_{gram_n} Count_{match}(gram_n)/ \\ &\qquad \sum_{S \in \{CandidateText\}} \sum_{gram_n} Count(gram_n) \end{aligned} \tag{23}$$

$$\begin{aligned} &ROUGE - N_{F-measure} \\ &\quad = 2 * ROUGE - N_{Recall} * ROUGE - N_{Precision}/ \\ &\qquad ROUGE - N_{Recall} + ROUGE - N_{Precision}, \end{aligned} \tag{24}$$

where n stands for the length of the n-gram $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate text and a reference text.

Both ROUGE-2 and ROUGE-SU4 compute the bigram recall, but ROUGE-SU4 also considers unigrams and skip-bigrams. Skip-bigram is a pair of words in the sentence order, allowing for gaps within a limited size which is always set to four.

When evaluating the slides, stop words are removed and stemming is utilized. We use two different ways to evaluate the slides based on the ROUGE toolkit. The first way is to evaluate and compare the whole generated slides and author-written slides, while the second one is to evaluate and compare the first 30 percentage, the middle 40 percentage and the last 30 percentage of both types of slides, respectively. If the generated slides can match the reference slides better in all parts, they should be considered to be better. We aim to better evaluate the coverage and structure of the generated slides by doing that. Paired T-Tests are applied between the ROUGE scores obtained in both evaluation ways between each baseline and our method. In the second evaluation way, we apply the T-Tests on the average ROUGE scores of the three parts.

Moreover, a user study is also performed to subjectively evaluate the slides generated by different methods. We randomly select 20 papers in the test set and employ TF-IDF method, MEAD and our method to generate slides. Random Walk and C-Lexrank are skipped because they are also summarization methods and get lower or similar performances as MEAD. These slides are presented to four human judges. The judges are students in computer science field. They do

TABLE 1
ROUGE F-Measure Scores when Evaluating the Whole Slides

Method	Rouge-1	Rouge-2	Rouge-SU4
TF-IDF	0.38859	0.11624	0.16424
Random Walk	0.39421	0.11555	0.16463
Mead	0.38778	0.11803	0.16239
C-Lexrank	0.38722	0.11223	0.15858
Our Method	**0.41342**	**0.13067**	**0.17502**

TABLE 2
ROUGE-1 F-Measure Scores when Evaluating the Segmented Slides

Method	Rouge-1			
	First 30%	Mid 40%	Last 30%	Avg
TF-IDF	0.28220	0.30732	0.28411	0.29121
Random Walk	0.29241	0.30661	0.28443	0.29448
Mead	**0.31132**	0.28063	0.25481	0.28225
C-Lexrank	0.28836	0.29834	0.27612	0.28761
Our Method	0.30235	**0.32662**	**0.29911**	**0.30936**

TABLE 3
ROUGE-2 F-Measure Scores when Evaluating the Segmented Slides

Method	Rouge-2			
	First 30%	Mid 40%	Last 30%	Avg
TF-IDF	0.07391	0.07713	0.07104	0.07402
Random Walk	0.07782	0.07822	0.06990	0.07531
Mead	**0.09023**	0.06664	0.06143	0.07277
C-lexrank	0.07722	0.07312	0.06862	0.07299
Our Method	0.08366	**0.08982**	**0.07920**	**0.08423**

TABLE 4
ROUGE-SU4 F-Measure Scores when Evaluating the Segmented Slides

Method	Rouge-SU4			
	First 30%	Mid 40%	Last 30%	Avg
TF-IDF	0.10586	0.11816	0.10100	0.10834
Random Walk	0.11052	0.11786	0.10048	0.10962
Mead	**0.11838**	0.10159	0.08967	0.10321
C-lexrank	0.10851	0.11048	0.09583	0.10494
Our Method	0.11357	**0.12780**	**0.10698**	**0.11612**

TABLE 5
T-Test p-Values between Each Baseline and Our Method when Evaluating the Whole Slides

System	T-Test		
	Rouge-1	Rouge-2	Rouge-SU4
TF-IDF	2.492E-13	2.749E-08	1.015E-06
Random Walk	5.329E-06	6.976E-11	1.125E-06
Mead	2.176E-11	1.601E-11	6.331E-08
C-lexrank	6.863E-22	9.143E-23	2.156E-22

TABLE 6
T-Test p-Values between Each Baseline and Our Method when Evaluating the Segmented Slides

System	T-Test		
	Rouge-1	Rouge-2	Rouge-SU4
TF-IDF	8.533E-14	1.557E-09	1.978E-07
Random Walk	2.841E-08	3.401E-09	1.085E-05
Mead	1.340E-17	1.758E-09	5.244E-13
C-Lexrank	6.893E-16	3.473E-11	7.876E-12

not know the identity of the system they are rating. They rate the slides based on subjects' own judgments. The judges are asked to answer the following questions by giving a rating on a scale of 1 to 5 for the presentations (5 means very good, 1 means very bad):

1) What is your satisfaction level of the slides' structure?
2) What is your satisfaction level of the slides' content?
3) What is your overall satisfaction level on the slides?

We also perform a factoid-based pyramid evaluation as in [17] on the paper set selected by the user study. Facts are annotated on the gold slides. Each fact in the gold slides is treated as a summarization content (SCU) [47]. The count of each fact's occurrences in the gold slides is regarded as the weight of this fact.

5.2 Results and Discussion

5.2.1 Comparison with Baseline Methods

The comparison results over ROUGE metrics are presented in Tables 1, 2, 3, 4. Table 1 shows that our proposed method can get better ROUGE scores, i.e., better content quality when evaluating the whole slides. It means that our generated slides are richer in content and much more similar to the human-written slides than those of the baselines as a whole.

Tables 2, 3 and 4 show that our method can get higher ROUGE scores on average when evaluating different parts of the slides. It means that the content texts of the slides generated by our method are more appropriately distributed over different sections. It demonstrates that the slides generated by our method have better structure than that generated by baseline methods.

Tables 5 and 6 show that the performance improvements of our method over baseline methods are statistically significant, because the T-Test p-values are very small.

As compared with the baseline methods, we obtain the importance scores of the sentences by learning from the human slides, which are deemed to be more credible. Moreover, the alignment between key phrases and sentences is also helpful to improve the content of the slides, because the sentences relevant to such key phrases can be considered to be more important. In our method, we distinguish global phrases from local phrases, and use local phrases to arrange contents in different sections, which makes the generated slides well-structured.

5.2.2 Comparison of Sentence Importance Computation Methods

Table 7 shows the comparison results of the different sentence importance computation methods. Note that the sentence scores by the methods are as input of our ILP model for slides generation.

We can see that the SVR-based methods (Overall-SVR, Avg-SVR, Our method) perform better than TF-IDF

TABLE 7
Comparison of Sentence Importance Computation Methods

Method	Rouge-1	Rouge-2	Rouge-SU4
Overall-SVR	0.40920	0.12853	0.17484
Avg-SVR	0.40988	0.12900	**0.17511**
TF-IDF	0.39962	0.12429	0.16926
Random Walk	0.40318	0.12721	0.17378
Our Method (Max-SVR)	**0.41342**	**0.13067**	0.17502

TABLE 8
ROUGE F-Measure Scores for Different Sentence Importance Training Models

Method	Rouge-1	Rouge-2	Rouge-SU4
SVM	0.40069	0.12136	0.16510
Linear Regression	0.40881	0.12558	0.16825
Our Method (Max-SVR)	**0.41342**	**0.13067**	**0.17502**

TABLE 9
ROUGE F-Measure Scores for Different ILP Models

Method	Rouge-1	Rouge-2	Rouge-SU4
SILP1	0.40629	0.12599	0.17177
SILP2	0.41288	0.12906	0.17295
Our Method	**0.41342**	**0.13067**	**0.17502**

and Random Walk. It proves that the SVR model can better estimate the importance scores of the sentences. The SVR model is trained from large dataset and the sentences scores predicted by the SVR-based method can be more reliable to be used for slides generation.

Among the three SVR-based methods, our method with the maximum similarity gets better ROUGE-1 and ROUGE-2 values than those with the overall similarity or the average similarity. Generally, slides can be divided into several parts and each part may be relevant to one section in the paper. The sentences in a specific section should be more similar to the corresponding part in the slides and less similar to the other parts. Using the maximum similarity can reflect this intuition well. So it is better to use the maximum similarity as the sentence importance scoring method.

5.2.3 Comparison of Sentence Importance Training Models

Table 8 shows the comparison results of the different sentence importance training models. The results indicate that regression models can get better performance than the classification model.

Both our SVR model and the linear regression model get better results than the SVM model. In our opinion, the real values predicted by the regression models can describe the sentence importance in a finer way than the binary numbers (i.e. 0, 1) predicted by the classification model. In addition, the real values fit the ILP model better than the binary numbers. The binary numbers cannot distinguish the differences of sentence importance in detail, but using the real numbers can obtain more information.

5.2.4 Comparison of ILP Models

Table 9 presents the comparison results between different ILP models. We can find that our ILP model gets better ROUGE scores than the two simplified ILP models.

Compared to SILP1, our adopted ILP model introduces the weights of the bigrams. The object function not only considers the count of the bigrams but also the weights of the bigrams. The author-written slides generally contain important sentences and important bigrams. By introducing the weights of the bigrams, the ILP solution tends to select more important bigrams and the generated slides can match the reference slides better and thus get better performance.

Compared to SLP2, our ILP model considers the scores of the covered sentences into the object function. In this way, the ILP solution tends to select bigrams and phrases that cover more important sentences. Then the bigrams and phrases selected can be more important. Therefore, it leads to an improvement of the slides' content quality.

5.2.5 Parameter Tuning

Fig. 3 presents the influences on the ROUGE scores when tuning the parameters λ_1, λ_2 and λ_3 of our method. We set the sum of the three parameters to one, and thus we actually need to change two of the three parameters. We can see that when the parameters are set in a wide range of values, our method can achieve high ROUGE scores. Our system generally performs better than the baselines. We can also see that all the three parts in our ILP model are helpful to get a better content quality for the generated slides. Without any part of them, the results will get worse.

5.2.6 User Study

Table 10 shows the average scores rated by human judgers for each method. The slides generated by our method obviously have better overall quality than the baseline methods. Being consistent with the automatic evaluation results, our slides are considered to have better content quality according to human judges. Moreover, owing to the indent structure and

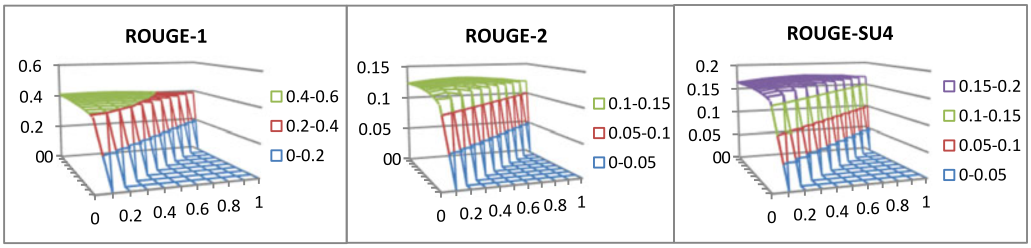


Fig. 3. Parameter influences (horizontal, vertical axis are λ_1, λ_2, respectively, $\lambda_3 = 1 - \lambda_1 - \lambda_2$).

TABLE 10
Average Rating Scores of Judges

Judge	Method	Structure	Content	Overall
1	TF-IDF	3.1	3.25	3
	Mead	2.6	3.1	2.85
	Our Method	3.6	3.8	3.65
2	TF-IDF	3.5	2.85	2.9
	Mead	3.2	3.45	2.8
	Our Method	3.6	3.5	3.45
3	TF-IDF	2.8	2.35	2.55
	Mead	2.5	1.4	1.7
	Our Method	3.75	3.35	3.7
4	TF-IDF	2.6	2.25	2.4
	Mead	2.5	2.15	2.2
	Our Method	3.8	3.15	3.5
avg	TF-IDF	3	2.68	2.71
	Mead	2.7	2.53	2.39
	Our Method	3.69	3.45	3.58

the alignment between phrases and sentences, the structure of our slides is also judged to be much better than the baselines' slides.

5.2.7 Pyramid Evaluation

Table 11 presents a sample fact list annotated. The facts are sorted by their occurrences in the gold slides. Table 12 shows the average factoid-based pyramid scores for each model. We can see our method can get better pyramid scores. We simultaneously extract key phrases and sentences in our model, and try to extract the most important key phrases. So the slides generated by our model can contain more important facts and get higher pyramid scores.

Overall, the experimental results indicate that our method can generate much better slides than the baselines in both automatic and human evaluations.

A few sample slides generated for one paper [48] are presented at the end of this paper. We can see that the slides are well-structured. People can quickly catch the key points when looking at the phrases and get more information after reading the sentences below the phrases. Using these draft slides, it could reduce much time for researchers when preparing their final presentations.

6 Conclusions and Future Work

This paper proposes a novel system called PPSGen to generate presentation slides from academic papers. We train a sentence scoring model based on SVR and use the ILP method to align and extract key phrases and sentences for generating the slides. Experimental results show that our method can generate much better slides than traditional methods.

TABLE 11
Sample Fact List Annotated

Fact	Occurrences
"Multi-hop wireless network"	5
"RTSCTS"	5
"Symmetric Incomplete State"	4
"Asymmetric Incomplete State"	4
"CSMACA protocol"	4
"Short-term unfairness"	4
. . .	. . .

TABLE 12
Average Pyramid Scores

Method	Pyramid score
TF-IDF	0.811
MEAD	0.746
Our Method	**0.863**

In future work, we will improve our system by using both text and graphical elements in the paper and make slides more comprehensible and vivid. When dealing with the graphical elements, we need to identify the graphical elements in the paper first. The relationship between the text elements and the graphical elements also needs to be identified. We need to know which sentences are most relevant to a graphical element and which graphical elements should be selected to generate the slides. We can use rule-based methods or machine learning based methods to solve the above problems. Then we can simply attach the tables and figures we select to the most relevant sentences in the slides.

In this paper, we only consider one typical style of slides that beginners usually use. In the future, we will consider more complicated styles of slides such as styles that slides are not aligned sequentially with the paper and styles that slides have more hierarchies. We will also try to extract the slide skeletons from the human-written slides and apply these slide skeletons to the automatic generated slides.

Furthermore, our system generates slides based on only one given paper. Additional information such as other relevant papers and the citation information can be used to improve the generated slides. We will consider this issue in the future.

Sample Slides

See on page 12.

Acknowledgments

The work was supported by National Natural Science Foundation of China (61170166, 61331011), Beijing Nova Program (2008B03) and National High-Tech R&D Program of China (2012AA011101). The authors thank the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

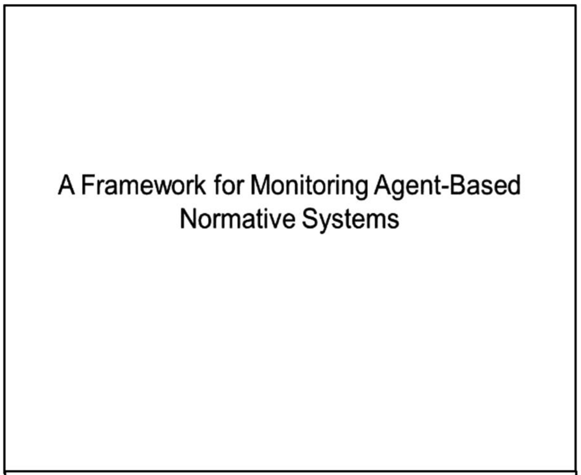
[1] M. Utiyama and K. Hasida, "Automatic slide presentation from semantically annotated documents," in *Proc. ACL Workshop Conf. Its Appl.*, 1999, pp. 25–30.

[2] Y. Yasumura, M. Takeichi, and K. Nitta, "A support system for making presentation slides," *Trans. Japanese Soc. Artif. Intell.*, vol. 18, pp. 212–220, 2003.

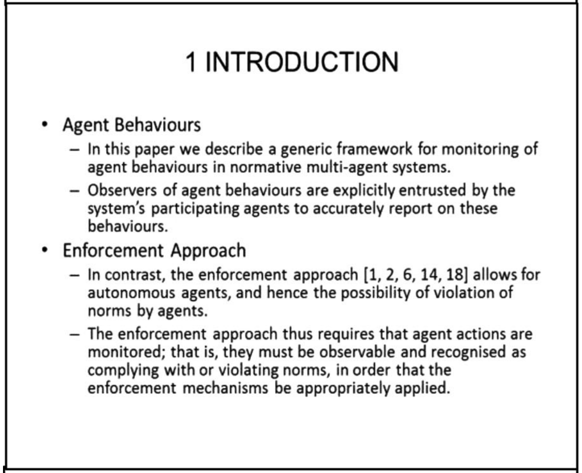
[3] T. Shibata and S. Kurohashi, "Automatic slide generation based on discourse structure analysis," in *Proc. Int. Joint Conf. Natural Lang. Process.*, 2005, pp. 754–766.

[4] T. Hayama, H. Nanba, and S. Kunifuji, "Alignment between a technical paper and presentation sheets using hidden Markov model," in *Proc. Int. Conf. Active Media Technol.*, 2005, pp. 102–106.

[5] M.Y. Kan, "SlideSeer: A digital library of aligned document and presentation pairs," in *Proc. 7th ACM/IEEE-CS Joint Conf. Digit. Libraries*, Jun. 2006, pp. 81–90.

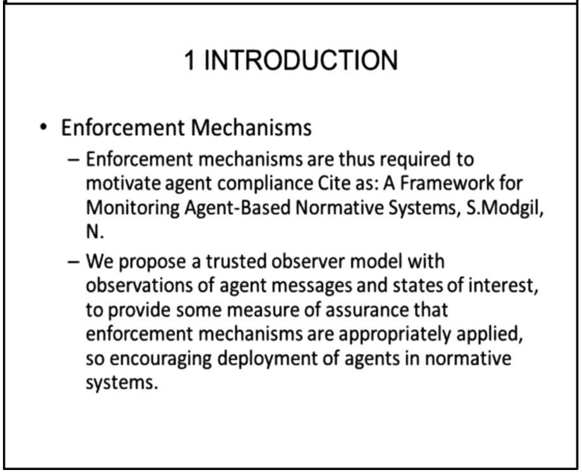


A Framework for Monitoring Agent-Based Normative Systems



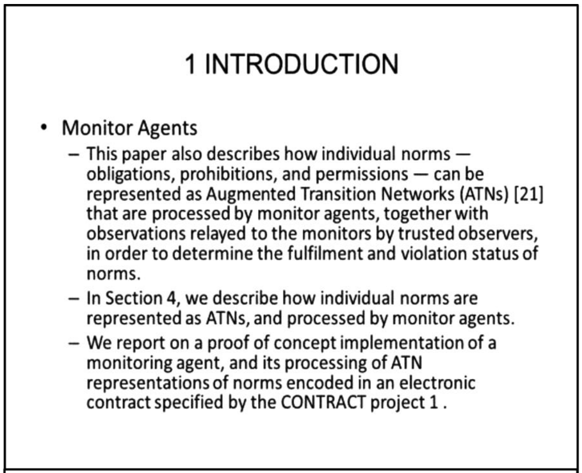
1 INTRODUCTION

- Agent Behaviours
  - In this paper we describe a generic framework for monitoring of agent behaviours in normative multi-agent systems.
  - Observers of agent behaviours are explicitly entrusted by the system's participating agents to accurately report on these behaviours.
- Enforcement Approach
  - In contrast, the enforcement approach [1, 2, 6, 14, 18] allows for autonomous agents, and hence the possibility of violation of norms by agents.
  - The enforcement approach thus requires that agent actions are monitored; that is, they must be observable and recognised as complying with or violating norms, in order that the enforcement mechanisms be appropriately applied.



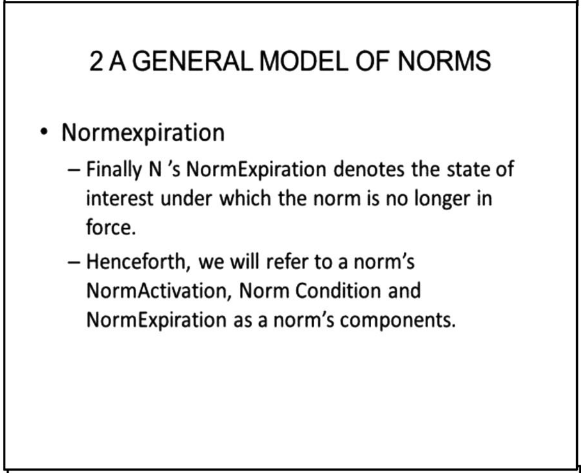
1 INTRODUCTION

- Enforcement Mechanisms
  - Enforcement mechanisms are thus required to motivate agent compliance Cite as: A Framework for Monitoring Agent-Based Normative Systems, S.Modgil, N.
  - We propose a trusted observer model with observations of agent messages and states of interest, to provide some measure of assurance that enforcement mechanisms are appropriately applied, so encouraging deployment of agents in normative systems.



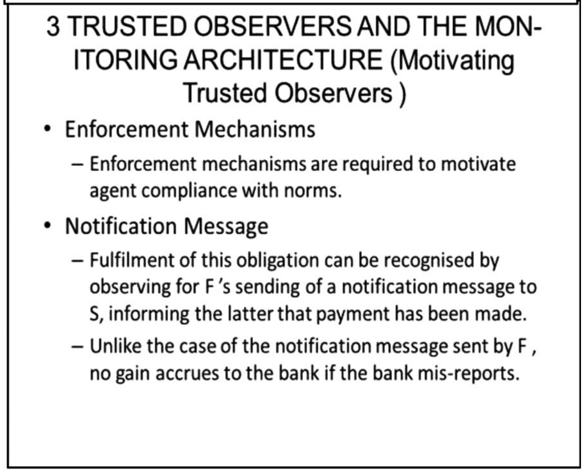
1 INTRODUCTION

- Monitor Agents
  - This paper also describes how individual norms — obligations, prohibitions, and permissions — can be represented as Augmented Transition Networks (ATNs) [21] that are processed by monitor agents, together with observations relayed to the monitors by trusted observers, in order to determine the fulfilment and violation status of norms.
  - In Section 4, we describe how individual norms are represented as ATNs, and processed by monitor agents.
  - We report on a proof of concept implementation of a monitoring agent, and its processing of ATN representations of norms encoded in an electronic contract specified by the CONTRACT project 1 .



2 A GENERAL MODEL OF NORMS

- Normexpiration
  - Finally N 's NormExpiration denotes the state of interest under which the norm is no longer in force.
  - Henceforth, we will refer to a norm's NormActivation, Norm Condition and NormExpiration as a norm's components.



3 TRUSTED OBSERVERS AND THE MON-ITORING ARCHITECTURE (Motivating Trusted Observers)

- Enforcement Mechanisms
  - Enforcement mechanisms are required to motivate agent compliance with norms.
- Notification Message
  - Fulfilment of this obligation can be recognised by observing for F 's sending of a notification message to S, informing the latter that payment has been made.
  - Unlike the case of the notification message sent by F , no gain accrues to the bank if the bank mis-reports.

[6] B. Beamer and R. Girju, "Investigating automatic alignment methods for slide generation from academic papers," in *Proc. 13th Conf. Comput. Natural Lang. Learn.*, Jun. 2009, pp. 111–119.

[7] S. M. A. Masum, M. Ishizuka, and M. T. Islam, "Auto-presentation: A multi-agent system for building automatic multi-modal presentation of a topic from world wide web information," in *Proc. IEEE/WIC/ACM Int. Conf. Intell. Agent Technol.*, 2005, pp. 246–249.

[8] S. M. A. Masum and M. Ishizuka, "Making topic specific report and multimodal presentation automatically by mining the web resources," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2006, pp. 240–246.

[9] M. Sravanthi, C. R. Chowdary, and P. S. Kumar, "SlidesGen: Automatic generation of presentation slides for a technical paper using summarization," in *Proc. 22nd Int. FLAIRS Conf.*, 2009, pp. 284–289.

[10] M. Sravanthi, C. R. Chowdary, and P. S. Kumar, "QueSTS: A query specific text summarization approach, in *Proc. 21st Int. FLAIRS Conf.*, 2008, pp. 219–224.

[11] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Develop.*, vol. 2, pp. 159–165, 1958.

[12] P. B. Baxendale, "Machine-made index for technical literature: an experiment," *IBM J. Res. Develop.*, vol. 2, no. 4, pp. 354–361, 1958.

[13] H. P. Edmundson, "New methods in automatic extracting," *J. ACM*, vol. 16, no. 2, pp. 264–285, 1969.

[14] A. Abu-Jbara and D. Radev, "Coherent citation-based summarization of scientific papers," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Lang. Technol.-Volume 1*, 2011, pp. 500–509.

[15] V. Qazvinian, D. R. Radev, S. M. Mohammad, B. J. Dorr, D. M. Zajic, M. Whidby, and T. Moon, "Generating extractive summaries of scientific paradigms," *J. Artif. Intell. Res.*, vol. 46, pp. 165–201, 2013.

[16] V. Qazvinian and D. R. Radev, "Identifying non-explicit citing sentences for citation-based summarization," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2010, pp. 555–564.

[17] V. Qazvinian and D. R. Radev, "Scientific paper summarization using citation summary networks," in *Proc. 22nd Int. Conf. Comput. Linguistics-Volume 1*, Aug. 2008, pp. 689–696.

[18] Q. Mei and C.Zhai, "Generating impact-based summaries for scientific literature," in *Proc. ACL*, vol. 8, pp. 816–824, 2008.

[19] M. A. Whidby, "Citation handling: Processing citation texts in scientific documents," Doctoral dissertation, Dept. Comput. Sci., Univ. Maryland, College Park, MD, USA, 2012.

[20] R. Jha, A. Abu-Jbara, and D. Radev, "A system for summarizing scientific topics starting from keywords," *ACM Comput. Surv*, vol. 40, no. 3, p. 8, 2013.

[21] S. Mohammad, B. Dorr, M. Egan, A. Hassan, P. Muthukrishan, V. Qazvinian, D. Radev, and D. Zajic, "Using citations to generate surveys of scientific paradigms," in *Proc. Human Lang. Technol.: The Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2009, pp. 584–592.

[22] P. Nakov, A. Schwartz, and M. Hearst, "Citation sentences for semantic analysis of bioscience text," in *Proc. SIGIR'04 Workshop Search Discovery Bioinformatics*, 2004, pp. 81–88.

[23] N. Agarwal, K. Gvr, R. S. Reddy, and C. P. Rosé, "Towards multi-document summarization of scientific articles: Making interesting comparisons with SciSumm," in *Proc. Workshop Autom. Summarization Different Genres, Media, Lang.*, 2011, pp. 8–15.

[24] O. Yeloglu, M. Evangelos, and Z.-H. Nur, "Multi-document summarization of scientific corpora," in *Proc. ACM Symp. Appl. Comput.*, 2011, pp. 252–258.

[25] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," in *Proc. ACL Workshop Intell. Scalable Text Summarization*, 1997, vol. 17, no. 1, pp. 10–17.

[26] D. Marcu, "From discourse structures to text summaries," in *Proc. ACL Workshop Intell. Scalable Text Summarization.*, 1997, vol. 97, pp. 82–88.

[27] I. Mani and E. Bloedorn, "Summarizing similarities and differences among related documents," *Inf. Retrieval*, vol. 1, no. 1, 2000, pp. 35–67.

[28] G. Erkan and D. R. Radev, "LexPageRank: Prestige in multi-document text summarization," in *Proc. EMNLP*, 2004, pp. 365–371.

[29] R. Mihalcea and P. Tarau, "A language independent algorithm for single and multiple document summarization," in *Proc. IJCNLP*, 2005, pp. 19–24.

[30] M. J. Conroy and D. P. O'leary, "Text summarization via hidden Markov models," in *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2001, pp. 406–407.

[31] D. Shen, J. T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, vol. 7, pp. 2862–2867.

[32] Y. Ouyang, S. Li, and W. Li, "Developing learning strategies for topic-based summarization," *Proc. 16th ACM Conf. Conf. Inf. Knowl. Manage.*, Nov. 2007, pp. 79–86.

[33] D. Galanis and P. Malakasiotis, "AUEB at TAC 2008," in *Proc. Text Anal. Conf.*, 2008.

[34] R. McDonald, "A study of global inference algorithms in multi-document summarization," in *Proc. Eur. Conf. Inf. Retrieval*, 2007, pp. 557–564.

[35] D. Gillick, B. Favre, and D. Hakkani-Tur, "The ICSI summarization system at TAC 2008," in *Proc. Text Anal. Conf.*, 2008.

[36] D. Gillick and B. Favre "A scalable global model for summarization," in *Proc. Workshop Integer Linear Program. Nat. Lang. Process.*, 2009, pp. 10–18.

[37] T. Berg-Kirkpatrick, D. Gillick, and D. Klein, "Jointly learning to extract and compress," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Lang. Technol.*, 2011, pp. 481–490.

[38] K. Woodsend and M. Lapata, "Multiple aspect summarization using integer linear programming," in *Proc. Joint Conf. Empirical Methods Nat. Lang. Process. Comput. Nat. Lang. Learn.*, 2012, pp. 233–243.

[39] D. Galanis, G. Lampouras, and I. Androutsopoulos, "Extractive multi-document summarization with integer linear programming and support vector regression," in *Proc. COLING*, 2012, pp. 911–926.

[40] V. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley, 1998.

[41] C. C. Chang and C. J. Lin. (2001), LIBSVM: A library for support vector machines, [Online]. Available: http: ((www.csie.ntu.edu.tw(∼cjlin(libsvm

[42] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang, "MEAD - A platform for multidocument multilingual text summarization," in *Proc. 4th Int. Conf. Lang. Resources Eval.*, 2004, pp. 1–4.

[43] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford Digital Libraries, Stanford, CA, USA, Tech. Report: SIDL-WP-1999-0120, 1999.

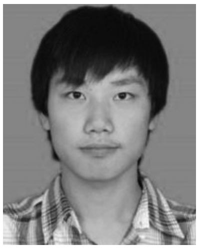
[44] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks." *Phys. Rev. E*, vol. 70, no. 6, p. 066111, 2004.

[45] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, no. 1, pp. 457–479, 2004.

[46] C. Y. Lin "ROUGE: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out, Post-Conf. Workshop ACL*, 2004, pp. 25–26.

[47] A. Nenkova and R. J. Passonneau, "Evaluating content selection in summarization: The pyramid method," in *HLT-NAACL*, vol. 4, pp. 145–152, May 2004.

[48] S. Modgil, N. Faci, F. Meneguzzi, N. Oren, S. Miles, and M. Luck, "A framework for monitoring agent-based normative systems," in *Proc. 8th Int. Conf. Auton. Agents Multiagent Syst.*, 2009, pp. 153–160.



Yue Hu received the BS degree from the Department of Computer Science and Technology of Peking University in 2012. He is currently a graduate student at the Institute of Computer Science and Technology of Peking University and is working with Prof. Xiaojun Wan. His research topic is scientific article summarization.

Xiaojun Wan received the BS, MS, and PhD degrees from Peking University in 2000, 2003, and 2006, respectively. He is currently a professor at the Institute of Computer Science and Technology of Peking University. His research interests include natural language processing and text mining. He has published more than 60 publications in major international conferences and journals, including ACL, SIGIR, AAAI, IJCAI, COLING, EMNLP, ICDM, CIKM, ACM TOIS, *Computational Linguistics*, *JASIST*, *KAIS*, *Information Sciences*, and so on. He was a PC member or area chair of major conferences such as ACL, SIGIR, EMNLP, COLING, CIKM, IJCNLP and NLPCC.