

SLIDE GENERATION APPROACH THROUGH CONTENT CATEGORIZATION

¹V.Meena, ²S.Mohana priya, ³V.Surya, ⁴G.Gnana Kumar

¹B.Tech-IT, ²B.Tech-IT, ³B.Tech-IT,

⁴Assistant Professor

¹Information Technology,

Jei Mathaajee College Of Engineering, Kanchipuram, Tamil Nadu

Abstract: *The proposed system is going to deal with a very challenging task of automatically generating presentation slides for academic papers. The wide availability of web documents in electronic forms requires an automatic technique to label the documents with a predefined set of topics, what is known as automatic Text Categorization (TC). Over the past decades, it has been witnessed a large number of advanced machine learning algorithms to address this challenging task. The presenter prepares their formal slides with the help of generated presentation slide in a quicker way. Documents are usually represented by the “bag-of-words”: namely, each word or phrase occurs in documents once or more times is considered as a feature. It first employs the regression method to learn the importance scores of the sentences in an academic paper, and then an effective algorithm is developed for multilabel classification with utilizing those data that are relevant to the targets. The key is the construction of a coefficient-based mapping between training and test instances, where the mapping relationship exploits the correlations among the instances, rather than the explicit relationship between the variables and the class labels of data and manufactures the multilevel classifier on the adapted low-dimensional information portrayals at the same time. It initially utilizes the relapse technique to take in the significance scores of the sentences in a scholastic paper, and after that adventures the Latent Dirichlet Allocation (LDA) strategy to create very much organized slides by choosing and adjusting key expressions and sentences to a point for the slide. We prepare a sentence scoring model in light of naïve Bayes classifier and utilize the LDA technique to adjust and remove key expressions and sentences for producing the slides. Exploratory outcomes demonstrate that our strategy can produce much preferred slides over conventional strategies.*

Index Terms - *l1-norm, instance-based learning-nearest neighbors (kNNs), multilabel classification, partial least square (PLS) regression*

I. INTRODUCTION

Slides have been an effective and popular means of presentation of information. In many conferences and meetings, a presenter takes the aid of slides to present his work in a systematic way (pictorial). In recent years with the availability of many software tools like Microsoft PowerPoint, Open office Presenter etc., for easy preparation of slides, their usage has increased tremendously. But these tools help only in the formatting of content (stylizing, bullet points etc.), but not in preparing the content itself. A user has to start from scratch and it is a time consuming task. In this work, we propose a tool that generates slides for the presentation with important points and all necessary figures, tables and graphs from a technical paper. As it is evident, such kind of a tool saves time and reduces the effort by providing a basic presentation, which can be further tuned/upgraded as final presentation. 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.

A presentation with slides is so effective to pass information to people in any situations, such as an academic conference or business. Although some software's, such as PowerPoint and Keynote, help us with making presentation slides, it is still cumbersome to make them from scratch. Slides contain the summarized version of a technical report. Given a document, “Automatic generation of presentation slides” becomes a nontrivial task because of challenges like segmental, Association for the Advancement of Artificial Intelligence.

A single page of a presentation is called slide. Collectively, a group of slides may be known as a slide deck. In the latter part of the 20th century, a presentation slide was created on a transparency and viewed with an overhead projector. Lecture slides in slide format are referred to as lecture notes and it is frequently downloadable by students in .ppt or .pdf format.

The slide rule, also known colloquially in the United States as slapstick, is a mechanical analog computer. For measuring length or drawing straight lines ordinary slide rules are not used.

Academic papers always have a similar structure. They generally contain several sections like abstract, introduction, related work, proposed method, experiments and conclusions. A presenter is also called as a beginner and the presentation slide can be written in various ways by different presenter. 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.

One of the most challenging tasks is automatic slides generation for academic papers. 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. One of the most difficult tasks is generally speaking, automatic slide generation 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 paper we concentrate on generating slides for research papers that are in accordance with standards of conference/journal proceedings. By and large, conference papers have an almost similar structure. They have an abstract and the sections present in them can be broadly classified into presenting the introduction, the related work, actual work (model), the

experiments, the conclusions and the bibliography. Most of the times, the presenter preserves the order of the paper in slides and each section is allotted one or more slides. A slide has a title and contains some bulleted points which are important in that section. Observing the similarity present between conference paper and human written slides for the paper, we address the problem of automatic generation of presentation slides by exploiting the structure of a conference paper. Here after we use terms “conference paper”, “technical paper”, “document” and “report” interchangeably.

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

II. BACKEND PROCESS

Net Beans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others. The Net Beans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM.

The Net Beans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the Net Beans Platform (including the Net Beans IDE itself) can be extended by third party developers. The Net Beans Team actively supports the product and seeks feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback.

Net Beans IDE provides first-class comprehensive support for the newest Java technologies and latest Java specification enhancements before other IDEs. It is the first free IDE providing support for JDK 8 previews, JDK 7, Java EE 7 including its related HTML5 enhancement and Java FX. With its constantly improving Java Editor, many rich features and an extensive range of tools, templates and samples, Net Beans IDE sets the standard for developing with cutting edge technologies out of the box. An IDE is much more than a text editor. The Net Beans Editor indents lines, matches words and brackets, and highlights source code syntactically and semantically. It also provides code templates, coding tips, and refactoring tools.

The editor supports many languages from Java, C/C++, XML and HTML, to PHP, Groovy, Java doc, JavaScript and JSP. Because the editor is extensible, you can plug in support for many other languages. Keeping a clear overview of large applications, with thousands of folders and files, and millions of lines of code, is a daunting task. Net Beans IDE provides different views of your data, from multiple project windows to helpful tools for setting up your applications and managing them efficiently, letting you drill down into your data quickly and easily, while giving you versioning tools via Subversion, Mercurial, and Git integration out of the box. When new developers join your project, they can understand the structure of your application because your code is well-organized. Net Beans provides static analysis tools, especially integration with the widely used Find Bugs tool, for identifying and fixing common problems in Java code. In addition, the Net Beans Debugger lets you place breakpoints in your source code, add field watches, step through your code, run into methods, take snapshots and monitor execution as it occurs.

The Net Beans Profiler provides expert assistance for optimizing your application's speed and memory usage, and makes it easier to build reliable and scalable Java SE, Java FX and Java EE applications. Net Beans IDE includes a visual debugger for Java SE applications, letting you debug user interfaces without looking into source code. Take GUI snapshots of your applications and click on user interface elements to jump back into the related source code. Net Beans IDE 7.0.1, which has full support for the official release of the Java platform.

III. ARCHITECTURE DIAGRAM

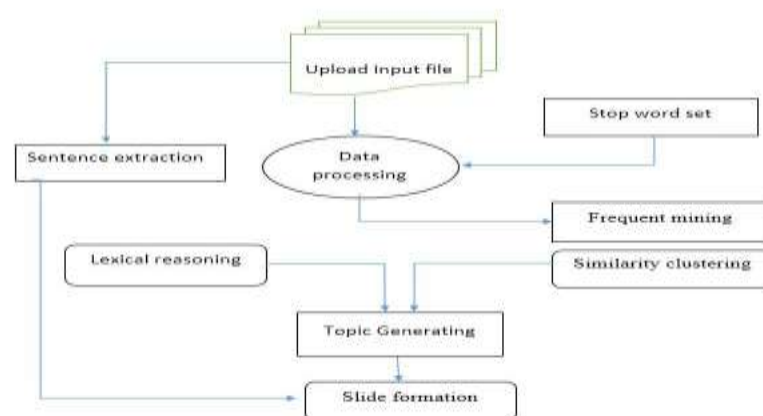


Fig.1 process of slide creation

IV. EXISTING SYSTEM

Most existing filter approaches first calculate class dependent feature scores, i.e., the feature importance for each class is measured. One major disadvantage is that using the combination operation may bias the feature importance for discrimination. They used four presentations from it to evaluate four aligners with the help of corpus of slide pairs and then it is to utilize methods such as TF-IDF term weighting and query expansion. The query does not improve performance in our application and that TF-IDF term weighting is inferior to a much simpler scoring mechanism based on the number of matched terms.

An inferior to a simpler scoring mechanism is TF-IDF term weighting and it is based only on the number of matched terms and query expansion degrades aligner performance. Our best aligner achieves an accuracy of 75%. To prepare the paper presentation they use much software such as Microsoft Power-Point and Open Office to help researchers prepare their slides.

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. A user has to start from scratch and it is a time consuming task

V. OUR WORK

The presenters prepare their formal slides with the help of generated presentation slide in a quicker way. A novel system is proposed to address this task. The importance scores of the sentences in an academic paper, and then exploits the LDA method to generate well-structured slides by selecting and aligning key phrases and sentences with the help of using the regression method.

We train a sentence scoring model based on naïve Bayes classifier and use the LDA 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 each sentence in a paper is learned by using the support vector regression (SVR) model. The presentation slides for the paper are generated by using the integer linear programming (ILP) model. The slides are generated automatically from the academic papers.

VI. CONCLUSION

We train a sentence scoring model based on naïve Bayes classifier and use the Latent Dirichlet Allocation 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. 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

VII. REFERENCES

- [1] Y. Xiao, B. Liu, Z. Hao, and L. Cao, "A similarity-based classification framework for multiple-instance learning," *IEEE Trans. Cyber.*, vol. 44, no. 4, pp. 500–515, Apr. 2014.
- [2] Y. Xu *et al.*, "Data uncertainty in face recognition," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1950–1961, Oct. 2014.
- [3] J. Zhang, X. Wu, and V. S. Sheng, "Active learning with imbalanced multiple noisy labeling," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1095–1107, May 2015.
- [4] M.-L. Zhang and Z.-H. Zhou, "Multi label neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.
- [5] J. Shen, Y. Zhao, S. Yan, and X. Li, "Exposure fusion using boosting Laplacian pyramid," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1579–1590, Sep. 2014.
- [6] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in Proc. IEEE Int. Conf. Data Min., Pisa, Italy, 2008, pp. 995–1000.
- [7] H. Liu, X. Li, and S. Zhang, "Learning instance correlation functions for multilabel classification," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 499–510, Feb. 2017.