Phrase-Based Presentation Slides Generation for Academic Papers

Sida Wang, Xiaojun Wan, Shikang Du
Institute of Computer Science and Technology, The MOE Key Laboratory of Computational Linguistics
Peking University, Beijing 100871, China
{sidawang, wanxiaojun, dusk}@pku.edu.cn

Abstract

Automatic generation of presentation slides for academic papers is a very challenging task. Previous methods for addressing this task are mainly based on document summarization techniques and they extract document sentences to form presentation slides, which are not well-structured and concise. In this study, we propose a phrase-based approach to generate well-structured and concise presentation slides for academic papers. Our approach first extracts phrases from the given paper, and then learns both the saliency of each phrase and the hierarchical relationship between a pair of phrases. Finally a greedy algorithm is used to select and align the salient phrases in order to form the well-structured presentation slides. Evaluation results on a real dataset verify the efficacy of our proposed approach.

Introduction

Presentation slides serve as a popular and effective means for researchers to present their research work on conferences, workshops or seminars. Researchers usually spend too much time on writing presentation slides for their published papers. Existing slides editors (e.g. Microsoft PowerPoint, OpenOffice, WPS Office and Latex) can help researchers to format the slides, but automatic generation of slides is still far from reach. It would be very useful to develop a tool to automatically generate draft slides by choosing and arranging text elements in an academic paper. The draft slides produced by the tool can be used as a basis by researchers when they prepare the final presentation slides, and thus save the researchers a great amount of time and effort.

The problem of automatic presentation slides generation is very challenging, because the slides usually have hierarchical structure consisting of bullet points in different levels. A bullet point can be a phrase, part of a sentence or a shortened sentence. In recent years, several pilot studies have investigated this challenging problem (Masao *et al.,* 1999; Yoshiaki *et al.,* 2003; Shibata and Kurohashi, 2005; Masum *et al.*, 2005; Masum and Ishizuka, 2006; Sravanthi *et al.*, 2009; Hu and Wan, 2013). Almost all of them adopt or adapt existing summarization techniques to extract sentences to form the presentation slides. Unfortunately, the slides produced in these studies are not well-structured and concise. For example, in the most recent work of (Hu and Wan, 2013), many long sentences are selected into the slides, and the selected key phrases are usually not good bullet points, making the slides boring to read.

In this study, we propose a phrase-based approach to address this challenging problem. Different from existing sentence-based approaches, our proposed approach considers phrases in the academic paper as the basic elements for content selection and arrangement. In particular, our approach first extracts phrases from the given paper, and then learns both the saliency score of each phrase and the hierarchical relationship between a pair of phrases. Finally a greedy algorithm is used to select and align the salient phrases in order to form the well-structured presentation slides. In this study, we propose two greedy algorithms for comparison. Empirical evaluation results on a real dataset verify the efficacy of our proposed phrase-based approach, which performs much better than the state-of-the-art PPSGen method and other baseline methods.

The rest of this paper is organized as follows: we first describe our proposed approach in details and then present the evaluation results. Finally, we briefly introduce related work and conclude this paper.

Our Approach

Though presentation slides can be written in different styles by different people, a typical type of presentation slides is bullet-based slides, and each bullet is usually a phrase, part of a sentence or a shortened sentence. Each slide has one or more first-level bullets (denoted as $L1$) and some first-level bullets may have several second-level bullets (denoted as $L2$). The second-level bullets are usually further explanations or specific descriptions of the corresponding first-level bullet. Sometimes, a second-level bullet may have several third-level bullets, and so on. Based on our empirical analysis of all bullets in a corpus of slides, we find that 56.2% bullets are first-level bullets, and 38.4% bullets are second-level bullets, and the first-level and second-level bullets cover 94.6% bullets in the slides. According to the above statistics, we ignore the third-level and subsequent lower-level bullets and focus on generating presentation slides with only two levels of bullets, as in (Hu and Wan, 2013).

In order to generate well-structured presentation slides for a given academic paper, our proposed approach relies on phrase selection and alignment, rather than sentence selection. Our phrase-based approach sequentially generate slides for each section in the given paper. It first extracts candidate phrases from each section with NLP tools, and then estimates the saliency of each candidate phrase and determines the hierarchical relationship of a phrase pair with machine learning techniques. Finally, it adopts greedy algorithms to select and align phrases to form the presentation slides. The details of the four steps are described as follows:

Candidate Phrase Extraction

We apply the Stanford parser[1] to parse sentences in the given paper. For each sentence, all NPs and the direct VPs under the sentence node (i.e. the root node) are extracted, referred to as candidate phrases. In this way, NPs in different granularities are extracted.

Phrase Saliency Estimation

This step aims to determine whether a candidate phrase should be selected into the slides, either in the first level or in the second level. The random forest classifier (Breiman 2001) is used to achieve this goal, and a classification model is trained based on the training set, which is built in the following way: we apply the Stanford parser to parse each bullet in the slides and extract NPs and VPs closest to the root of the parse tree, and then match the candidate phrases in the paper and the gold-standard phrases in the slides. If the cosine similarity between a candidate phrase and a gold-standard phrase is larger than 0.5, the two phrases are matched. The candidate phrases which have at least one match with the gold-standard phrases are treated as positive examples, and the others are treated as negative examples.

We extract the following features for each phrase p:

Phrase position: This group of features include the paragraph position of the phrase in a section, the sentence position of the phrase in a section and the sentence position of the phrase in a paragraph. For example, if the paragraph of the phrase is the n^{th} paragraph in a section, then its paragraph position is n.

Phrase length: It refers to the number of words in the phrase after stop words in the phrase are removed.

Tf-idf: The features include average term frequencies/tf-idf values of unigrams/bigrams in the phrase.

Section: It indicates whether the section in which p appears is Abstract, Introduction, Related Work, Method, Evaluation or Conclusion.

Phrase type: It refers to phrase type of the phrase, with 1 for VP and 0 for NP.

Parse tree information: The features include the depth and height of the phrase in the parse tree. Depth denotes the number of nodes on the path from the phrase to the root of the parse tree. Height denotes the maximum number of nodes on the path from the phrase to the leaves of the parse tree under the phrase.

We choose the random forest classifier because it generally outperforms other classifiers based on our empirical analysis. The Scikit-learn toolkit[2] is used and the probability of prediction is acquired through the API function of predict_proba. The output probability of the classifier is used for measuring the saliency of a phrase. Only phrases with probability higher than 0.5 are kept.

We further remove redundant phrases in the candidate set in the following way: 1) If both phrase p and the descendants of p in the parse tree are in the candidate set, the descendants of p are removed and only p is kept. 2) If the cosine similarity between two phrases is larger than 0.8, then the phrase with higher saliency is kept and the other phrase is removed.

Phrase Relationship Prediction

This step aims to determine the hierarchical relationship (i.e. the L1-L2 relationship) between two phrases. In particular, given a pair of candidate phrases $< p, q >$, it aims to predict whether phrase p should be put on the upper level of phrase q. We also employ the random forest classifier to achieve this goal, and the classifier is used to classify whether each phrase pair has hierarchical relationship or not. The training set is built as follows: we extract the pairs of gold-standard phrases which have hierarchical relationships in the slides, and then match the candidate phrases in the paper with the gold-standard phrases, and use the matched phrase pairs as positive examples. A random sample of other phrase pairs is used as negative examples.

[1] http://nlp.stanford.edu/software/lex-parser.shtml

[2] http://scikit-learn.org/stable/index.html

In order to describe the features concisely, we introduce the definition of actual difference and relative difference. Given a pair of phrases $< p, q >$ and the feature of p as $f(p)$, the feature of q as $f(q)$, the actual difference is computed as:

$$AD(< p, q >) = f(p) - f(q)$$

The relative difference is computed as:

$$RD(< p, q >) = \frac{f(p) - f(q)}{f(p)}$$

In the case $f(p) = 0$, we let $RD(< p, q >) = 0$.

Give a pair of phrases $< p, q >$, we extract and use all the features mentioned above in previous section for p and q. In addition to these features, we derive the following new features:

Difference of position: The features include the actual difference between the paragraph positions of the two phrases in sections, the sentence positions of the two phrases in sections and the sentence positions of the two phrases in paragraphs.

Difference of phrase length: The features include the actual difference and relative difference between phrase lengths of p and q.

Difference of tf-idf: The features include the actual difference and relative difference between the average term frequencies/ tf-idf values of unigrams/bigrams of p and q.

Difference of parse tree information: The features include the actual difference and relative difference between the depths and heights in the parse trees of p and q.

Difference of phrase type: It indicates whether p and q are both noun phrases/verb phrases or not.

In order to address the class imbalance problem, the numbers of positive examples and negative examples are resampled to 1:1 for learning.

Finally, a probability value is acquired for each phrase pair by the Scikit-learn toolkit, indicating how strong the L1-L2 relationship of the phrase pair is. Only the relationships with probability higher than 0.5 are kept.

Note that the predicted relationships between candidate phrases form a directed graph G. In G, a phrase A could be put in the lower level of B and in the meantime put in the upper level of C. What's more, B could be in the upper level of C and in some cases, even circles exist, which makes the situation very complex.

Phrase Selection and Alignment

This step aims to determine which candidate phrases are finally selected and used in the slides. We have two objectives in this final step: the first is to select as many phrases with high saliency as possible given the length limit of the slides; the second is to align as many pairs of phrases with strong hierarchical relationships as possible to generate the hierarchical bullets. The two objectives are hard to be satisfied at the same time and the optimization problem is intractable. Therefore, we propose two greedy algorithms to find approximate solutions to the problem and generate slides accordingly.

Algorithm 1: Relationship first algorithm

```
Input : Set of candidate phrase pairs with hierarchical relationships
        CP, set of candidate phrases C, length limit L
Output: Set of selected phrase pairs S
U1 = ∅   // initialize the set of selected first-level phrases
U2 = ∅   // initialize the set of selected second-level phrases
S = ∅            // initialize the set of selected phrase pairs
curlen = 0              // initialize number of words in S
while CP ≠ ∅ do
    select phrase pair <p, q> ∈ CP with the highest probability
    if p ∈ U2 or q ∈ U1 then
        do nothing
    else
        if p ∈ S then
            totlen = len(q)          // len(q): number of words of q
        else
            totlen = len(p) + len(q)
        end
        if curlen + totlen ≤ L then
            S = S ∪ {<p, q>}
            U1 = U1 ∪ {p}
            U2 = U2 ∪ {q}
            curlen = curlen + totlen
        end
    end
    CP = CP \ {<p, q>}
    C = C \ {p, q}
end
while C ≠ ∅ do
    select phrase p ∈ C with the highest saliency probability
    if curlen + len(p) ≤ L then
        S = S ∪ {<p, NULL>}          // NULL means p has no L2
        curlen = curlen + len(p)
    end
    C = C \ {p}
end
```

Before running the following greedy algorithms, we perform phrase merging in the following way: if a noun phrase with saliency probability higher than 0.5 and a verb phrase with saliency probability higher than 0.5 are under the same root tag in the parse tree of a sentence, they are merged into a sentence and the sentence's saliency score and the hierarchical relationship information are inherited from the verb phrase. In this way, we can keep some informative sentences in the generated slides.

Relationship First Algorithm (Relation-First)

In this algorithm, we prefer selecting phrase pairs with strong hierarchical relationships to selecting individual phrases with high saliency. The algorithm iteratively find and settle the most reliable relationship. This means that given the phrase pair $< p, q >$ with the highest predicted probability, p is confirmed as $L1$ and q as $L2$. Any subsequent relationship conflicting with the established one is abandoned, i.e., if pair $< p, q >$ is confirmed, neither $< e, p >$ nor $< q, d >$ are eligible. The details of the algorithm is illustrated in Algorithm 1.

Saliency First Algorithm (Saliency-First)

In this algorithm, we prefer selecting individual phrases with high saliency. Given the directed graph G, we assume that the vertex with indegree of 0 is $L1$, and the adjacent nodes of $L1$ are defined as $L2$. With this definition, we iteratively choose the phrases with highest saliency probability. If the selected phrase is $L1$, then it is selected, and if it is $L2$, then the most probable parent phrase is selected as well. The procedure is repeated until the length limit is met. The details of the algorithm is illustrated in Algorithm 2.

Algorithm 2: Saliency first algorithm

```
Input  : Set of candidate phrases C, set of first-level phrases U1,
         set of second-level phrases U2, length limit L
Output: Set of selected phrase pairs S
S = ∅                    // initialize the set of selected phrase pairs
P = ∅                    // initialize the set of selected phrases
curlen = 0               // initialize number of words in S
while C ≠ ∅ do
    select phrase p with the highest saliency probability
    if p ∈ U1 then
        if curlen + len(p) ≤ L then
            P = P ∪ {p}
            S = S ∪ {<p, NULL>}      // NULL means p has no L2
            curlen = curlen + len(p)
        end
    else if p ∈ U2 then
        select the most probable parent q from U1
        if q ∈ P then
            totlen = len(p)
        else
            totlen = len(p) + len(q)
        end
        if curlen + totlen ≤ L then
            P = P ∪ {q, p}
            S = S ∪ {<q, p>}
            curlen = curlen + totlen
        end
    end
    C = C \ {p, q}
end
```

Evaluation

Data Set

We randomly collected 175 pairs of paper and slides in the computer science field in the same way as in (Hu and Wan 2013). We adopted the same preprocessing steps to extract texts and detect physical structures from papers and slides. Due to poor recognition of bullet relationship with automatic tools, we manually corrected the bullet relationship in slides. In our experiments, 100 pairs of paper and slides are used for training, 25 for validation and 50 for testing.

Evaluation Metric

Former evaluation metrics treat the whole slides as a single summary text for evaluation and neglects the bullet structure. In order to evaluate the bullet structure of the slides at a finer granularity, we design a new metric for this special task.

The basic idea is treating each $L1$ bullet with all its $L2$ bullets as a cluster, and comparing the set of clusters (denoted as C_{gen}) in generated slides with the set of clusters (denoted as C_{gold}) in gold-standard slides:

$$S(C_{gen}, C_{gold}) = \frac{1}{|C_{gold}|} \sum_{g \in C_{gold}} argmax_{h \in C_{gen}} sim(g, h)$$

where *sim(g, h)* denotes the matching score between two bullet clusters g and h, and is computed by separately matching the bullets of different levels in g and h. The matching between bullets is measured with the popular ROUGE-N metrics (Lin & Hovy, 2003).

Specifically, given a cluster m, we define the $L1$ of the cluster as $L1_m$, and the set of $L2$ as U_m. Then the matching score between two bullet clusters g and h are computed as follows:

$$sim(g,h) = w_1 * s(L1_g, L1_h) + w_2 * s(U_g, U_h)$$

$$s(L1_g, L1_h) = ROUGE - N_{F1}(L1_g, L1_h)$$

$$s(U_g, U_h) = mean_{u \in U_g} argmax_{u \in U_h} ROUGE\text{-}N_{F1}(u, v)$$

where w_1 and w_2 are different weights associated with the L1 part and the L2 part. Based on different settings of the weights, we use the following three metrics in the experiments:

$$(w_1, w_2) = \begin{cases} \left(\frac{1}{2}, \frac{1}{2}\right), metric - A \\ \left(\frac{2}{3}, \frac{1}{3}\right), metric - B \\ \left(\frac{1}{|U_g| + 1}, \frac{|U_g|}{|U_g| + 1}\right), metric - C \end{cases}$$

The above evaluation metrics borrow the same idea of evaluating data clustering results (Karypis et al., 2000). Given each bullet cluster g in the reference slides, scores are calculated between g and every bullet cluster h in the generated slides, and the maximum score is taken as the score of g. In other words, we find a best matched bullet cluster for each gold-standard bullet cluster. Finally, scores of all clusters in C_{gold} are averaged as the overall evaluation score for the whole slides. F1 measure of $ROUGE\text{-}N$ is used for measuring two single bullets based on n-gram overlap with stop words removed.

Evaluation Results

Our proposed phrase-based approaches with two greedy algorithms (**Saliency-First** and **Relation-First**) are compared with the following methods:

LexRank (Erkan and Radev, 2004): It is a graph-based method computing eigenvalue centrality as the saliency score of sentence. The cosine similarities between tf-idf vectors of sentences are used as the weights of edges in the graph.

LSA (Steinberger and Jezek, 2004): Singular Value Decomposition (SVD) is applied to find principal and mutually orthogonal dimensions of sentence vectors, and a representative sentence is picked out from each of the dimensions.

SumBasic (Nenkova and Vanderwende, 2005): It is a sentence extraction method based on average probability of words in the sentence. After the best scoring sentence is chosen, probabilities of words in the sentence are updated.

Luhn (Luhn 1958): It is a heuristic approach based on word frequency to select sentences.

PPSGen (Hu and Wan, 2013): It is the state-of-the-art method for presentation slides generation. The method adopts machine learning techniques to determine which sentences are salient. The sentences and phrases are selected and aligned in an ILP-based optimization framework.

The length of generated slides is limited to 15% of the text length (i.e., number of words) in the given paper. Since methods like LSA, TextRank, SumBasic and Luhn cannot provide hierarchical structures as our methods and PPSGen do, the extracted sentences by these methods are treated as L1.

	metric-A		metric-B		metric-C	
n-gram	1	2	1	2	1	2
Relation-First	**0.26**	**0.101**	**0.28**	**0.109**	**0.29**	**0.117**
Saliency-First	0.25	0.097	0.27	0.105	0.28	0.113
PPSGen	0.14	0.035	0.14	0.041	0.15	0.043
LexRank	0.12	0.038	0.13	0.040	0.14	0.042
SumBasic	0.12	0.037	0.13	0.038	0.14	0.041
Luhn	0.11	0.036	0.12	0.038	0.12	0.041
LSA	0.10	0.028	0.11	0.030	0.12	0.032

Table 1. Comparison results of different methods

n_gram	1	2
Relation-First	**0.30**	**0.117**
Saliency-First	0.29	0.114
PPSGen	0.15	0.046
LexRank	0.17	0.053
LSA	0.15	0.042
Luhn	0.16	0.053
SumBasic	0.16	0.050

Table 2. Comparison results without differentiating L1 and L2 bullets

Table 1 shows the comparison results of different methods. Results show that methods generating hierarchical bullet structures (i.e. Saliency-First, Relation-First and PPSGen) generally outperform other methods, except for the ROUGE-2 scores on metric-A for PPSGen. More importantly, our proposed approaches perform much better than other methods over different metrics, including the state-of-the-art PPSGen method. The reason is that our generated bullet structures can well match the bullet structures in the gold-standard slides, while PPSGen relies mainly on sentence extraction and cannot generated well-structured slides. We can also see that the Relation-First method performs slightly better than the Saliency-First method. The comparison results demonstrate the very promising future for phrase-based methods for this challenging task.

Since most baseline methods do not generate the L1-L2 hierarchical structures in the slides, we further evaluate the generated slides without differentiating the L1 and L2 bullets in each cluster, i.e., treating each bullet cluster as a single text for comparison. In other words, we ignore the L1-L2 structure and concatenate the text in L1 and all the texts in L2 into a single text, and then compute the matching score *sim(g, h)* based on the ROUGE-N metrics. The comparison results are shown in Table 2. We can still find that our proposed methods perform much better than other methods. Similarly, the Relation-First method performs slightly better than the Saliency-First method, which further demonstrates the importance and usefulness of the hierarchical relationships between phrases.

Running Example

In this section, we show some example slides generated by different methods for an example paper ("Rethinking Data Management for Storage-centric Sensor Networks") in Figures 1~3. Two example slides are presented for Relation-First, Saliency-First and PPSGen methods, respectively.

We can see that the slides generated by PPSGen are boring to read, because the method mainly selects very long sentences into the slides, and the generated slides are lengthy. Another shortcoming of the slides generated by PPSGen is that each L1 bullet is often a single word chosen from the lower-level sentence, but the word is not very informative and representative.

As compared with the slides generated by PPSGen, the slides generated by our methods are concise and easier to read because the slides are mainly composed of phrases instead of long sentences. The L1-L2 hierarchical structure is not required for each L1, and thus some L1 bullets have corresponding L2 bullets, while other L1 bullets do not. For example, the slides generated by Relation-First summarize several challenges that arise in the design of stonesDB by using several L2 bullets. We can also see some short sentence can be formed and used by phrase merging in the generated slides. In all, the bullet structures of our generated slides are better than that of PPSGen.

Lastly, we can see that the slides generated by Relation-First adopt more L1-L2 relations than that generated by Saliency-First, which is in accordance with our expectations since the former method preferably considers phrase pairs with strong hierarchical relationships.

Related Work

Automatic slides generation for academic papers remains far under-investigated and there exist only several pilot studies investigating this task (Masao et al., 1999; Yoshiaki et al., 2003; Shibata and Kurohashi, 2005; Masum et al., 2005; Masum and Ishizuka, 2006; Sravanthi et al., 2009; Hu and Wan, 2013). Almost all of them adopt or adapt document summarization techniques to select sentences or clauses to form the slides. Among these studies, the state-of-the-art work is the PPSGen method (Hu and Wan, 2013). However, PPSGen mainly extracts sentences and the bullet structures are not well constructed.

Other related works include poster generation (Qiang et al., 2016), paper-slide alignment (Hayama et al., 2005; Kan 2006; Beamer and Girju, 2009) and phrase-based or segment-based document summarization (Chuang and Yang, 2000; Bing et al., 2015; Yao et al., 2015).

Conclusion and Future Work

In this paper, we propose a phrase-based approach and implement two greedy algorithms to generate well-structured and concise presentation slides for papers. Evaluation results on a real dataset verify the efficacy of our proposed methods. In future work, we will collect more data and try deep learning techniques for better estimating phrase saliency and predicting the relationship between phrases. We will also try to add equations, figures and tables into the generated slides to make the slides more attractive.

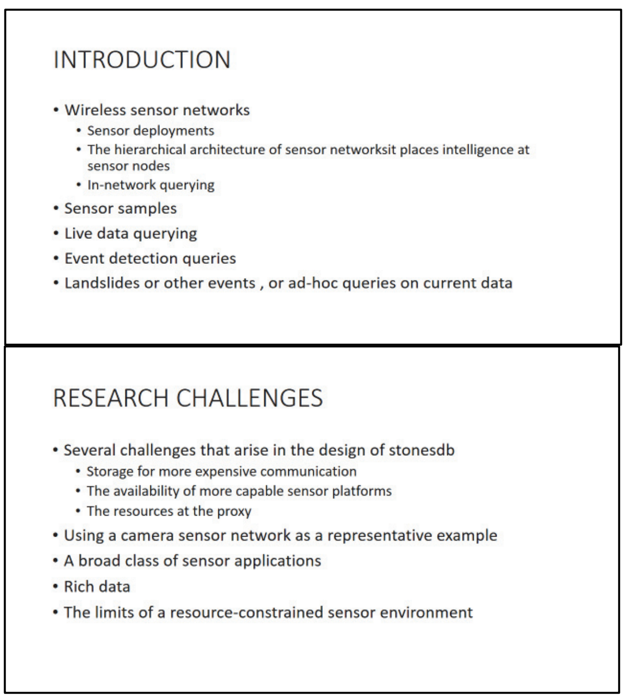

Figure 1. Example slides generated by Relation-First

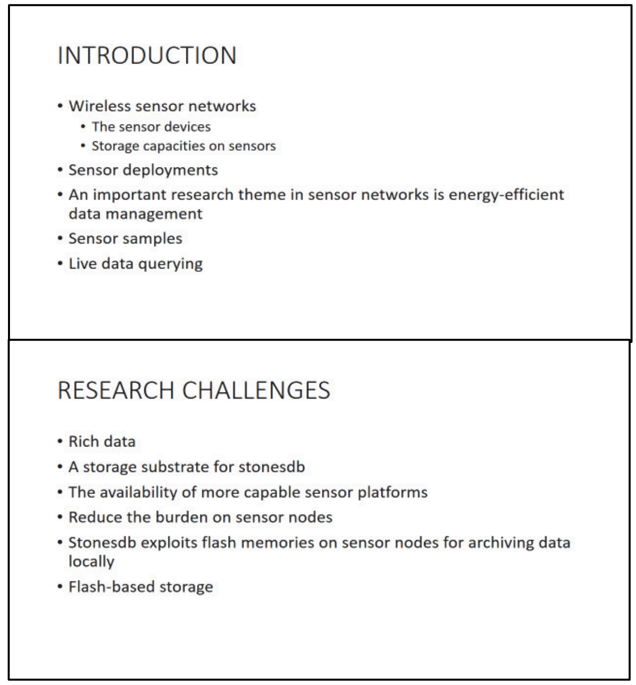

Figure 2. Example slides generated by Saliency-First

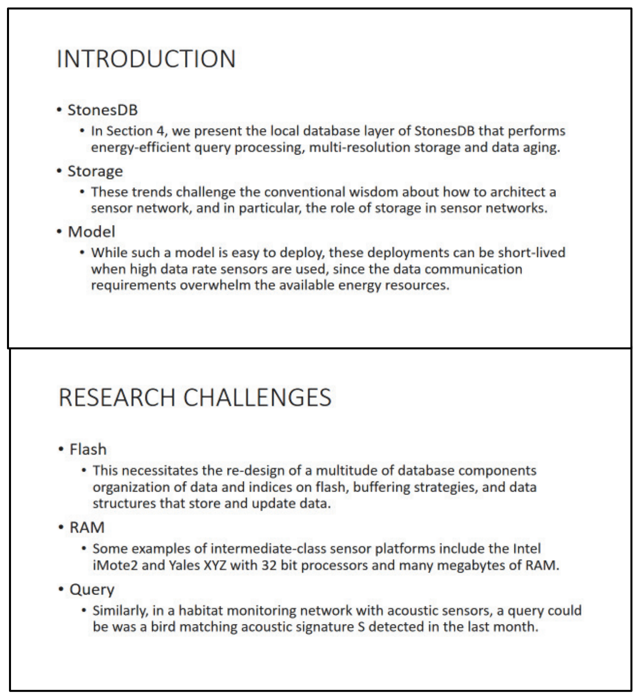

Figure 3. Example slides generated by PPSGen

Acknowledgments

This work was supported by NSFC (61331011), 863 Program of China (2015AA015403) and IBM Global Faculty Award Program. Xiaojun Wan is the corresponding author. Sida Wang and Shikang Du are also affiliated with School of Physics, Peking University.

References

Beamer, B. and Girju, R. 2009. Investigating Automatic Alignment Methods for Slide Generation from Academic Papers. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 111–119.

Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., and Passonneau, R. J. 2015. Abstractive multi-document summarization via phrase selection and merging. *Proceedings of ACL-2015*.

Breiman, L., 2001. Random forests. *Machine learning*, *45(1)*:5-32.

Chuang, W. T., and Yang, J. 2000. Extracting sentence segments for text summarization: a machine learning approach. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 152-159). ACM.

Erkan, G., and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. (JAIR)* 22:457–479.

Hovy, E., Lin, C.Y., Zhou, L. and Fukumoto, J., 2006. Automated summarization evaluation with basic elements. In *Proceedings of the Fifth Conference on Language Resources and Evaluation*, *LREC 2006*, 604-611.

Hayama, T., Nanba, H., Kunifuji, S. 2005. Alignment between a technical paper and presentation sheets using hidden markov model. In *Proceedings of the 2005 International Conference on Active Media Technology*.

Hu, Y. and Wan, X., 2013. PPSGen: Learning to Generate Presentation Slides for Academic Papers. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, *IJCAI*, 2013, 2099-2015.

Kan, M.-Y. 2006. SlideSeer: A digital library of aligned document and presentation pairs. *JCDL'06*, June 11–15, Chapel Hill, North Carolina, USA.

Karypis, M. S. G., Kumar, V., & Steinbach, M. 2000, August. A comparison of document clustering techniques. *In TextMining Workshop at KDD2000*.

Lin, C. Y. and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 71-78). Association for Computational Linguistics.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development* 2(2):159–165.

Masao, U. and Kôiti, H., 1999. Automatic slide presentation from semantically annotated documents. In *Proceedings of the Workshop on Coreference and its Applications*. Association for Computational Linguistics.

Masum, S.M., Ishizuka, M. and Islam, M.T., 2005. 'Auto-Presentation': a multi-agent system for building automatic multi-modal presentation of a topic from World Wide Web information. In *Proceedings of the European Conference on Information Retrieval* 2005, 557–564

Masum, S.M.A. and Ishizuka, M., 2006. Making topic-specific report and multimodal presentation automatically by mining the web resources. In *Proceedings of the 2006 International Conference on Web Intelligence*, 240-246.

Nenkova, A., and Vanderwende, L. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*.

Qiang, Y., Fu, Y., Guo, Y., Zhou, Z. H., & Sigal, L. 2016. Learning to Generate Posters of Scientific Papers. In *Proceedings of AAAI-2016*.

Shibata, T. and Kurohashi, S., 2005. Automatic Slide Generation Based on Discourse Structure Analysis. *Natural Language Processing–IJCNLP 2005*, 754-766.

Sravanthi, M., Chowdary, C.R. and Kumar, P.S., 2009. SlidesGen: Automatic Generation of Presentation Slides for a Technical Paper Using Summarization. In *Proceedings of the Twenty-Second International FLAIRS Conference* 2009, 284-289.

Steinberger, J., and Jezek, K. 2004. Text summarization and singular value decomposition. In *Advances in Information Systems, Third International Conference*, *ADVIS 2004*, 245– 254.

Vanderwende, L. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*.

Yao, J. G., Wan, X., and Xiao, J. 2015. Phrase-based compressive cross-language summarization. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 118-127).

Yoshiaki, Y., Masashi, T., Katsumi, N. A support system for making presentation slides (in Japanese). 2003. *Transactions of the Japanese Society for Artificial Intelligence*. 18: 212-220.