Distributed Keyword Vector Representation for Document Categorization

Yu-Lun Hsieh, Shih-Hung Liu,
Yung-Chun Chang, Wen-Lian Hsu

Institute of Information Science, Academia Sinica, Taiwan
morphe@iis.sinica.edu.tw

Outline

- Introduction
- Previous Work
- Proposed Method
- Experiments
- Results & Discussion
- Conclusion

Introduction

- How to quickly categorize huge amount of text has become a challenging problem in the modern age
- By means of current computational technologies, we can quickly collect and classify the topic of a news document
- Individuals and businesses can both benefit from this to find documents of their interests

Topic as Category

- A topic is essentially associated with specific times, places, and persons (Nallapati et al., 2004)
- These terms can be considered as keywords, and utilized for classification purposes.
- In this work, we examine the power of neural-network based representations in capturing the relations between those keywords on the surface, and the topic of the document.

Outline

- Introduction
- Previous Work
- Proposed Method
- Experiments
- Results & Discussion
- Conclusion

Previous Work

- Most previous methods rely on some measures of the importance of <u>keyword</u> features
  - Keyword weighting based on traditional statistical methods such as TF*IDF, conditional probability, and/or generation probability
  - It has been proven that keywords are very important in text categorization tasks

Previous Work (II)

- Machine learning approaches:
  - Supervised: given a training corpus containing a set of manually-tagged examples of predefined topics, a supervised classifier is employed to train a topic detection model to classify a document
  - Unsupervised: clustering of keywords and/or semantic information in text

Text Representation

- A document can be represented as a vector for the computer to learn a classifier
  - e.g., vector space model, SVMs, kNN, and logistic regression
- Or, use latent semantic information to model the relationships between text and its topic
  - e.g., latent semantic analysis (LSA), probabilistic LSA, and latent Dirichlet allocation (LDA)

Neural Network

- Recently, there is an exploding interest in representing words or documents through neural network (NN), or ‘deep learning’ models
- It inspired us to use vectors learned from NNs and a robust vector-based classifier to categorize text
- Utilize the power of NNs to capture hidden connections between words and topics

Outline

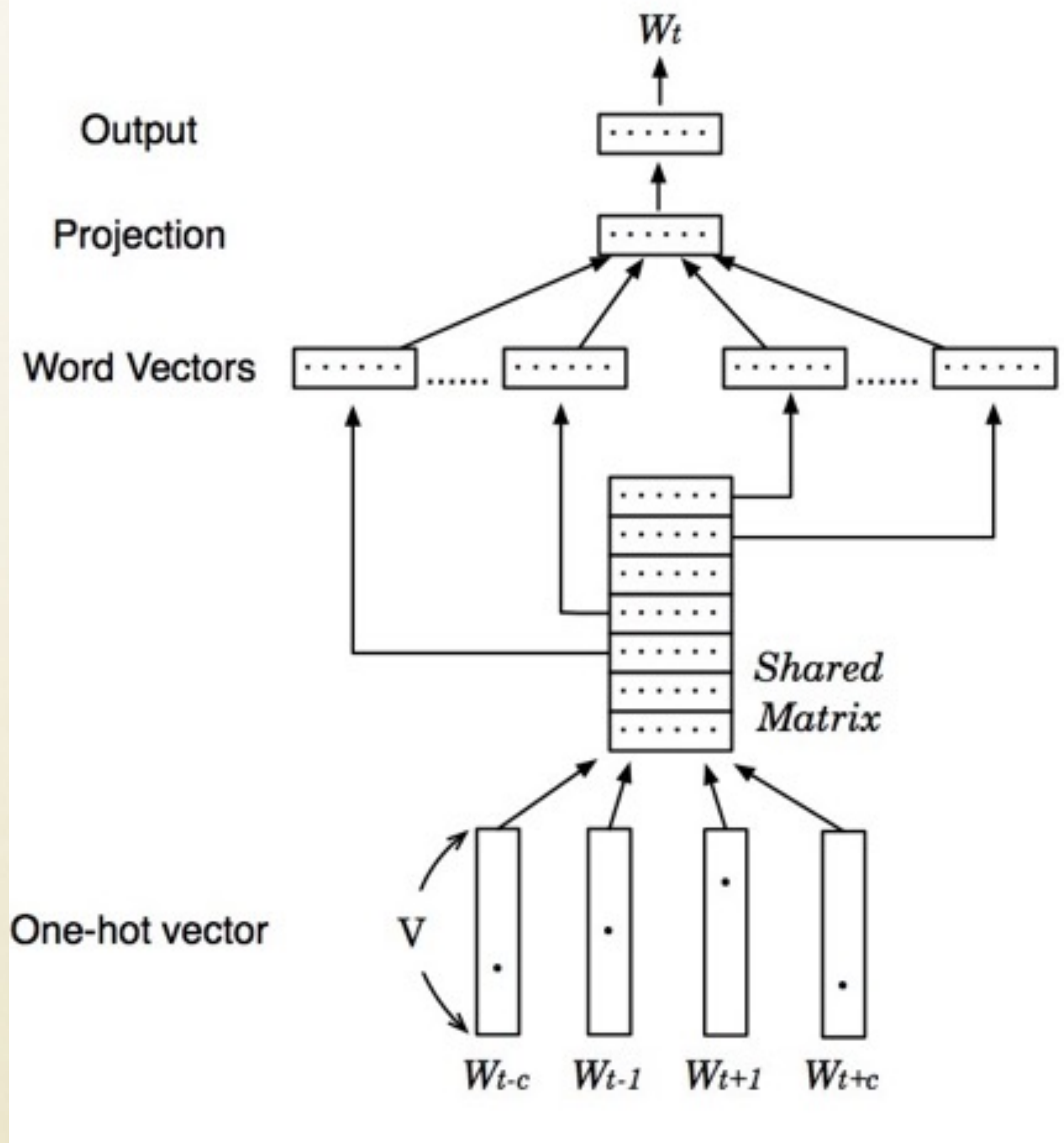
- Introduction
- Previous Work
- Proposed Method
- Experiments
- Results & Discussion
- Conclusion

Method

- We propose a novel use of word embedding for text classification
  - Word embedding: a by-product of neural network language model
- It can learn hidden semantic and syntactic regularities in various NLP applications
- Representative methods for the word level include the continuous bag-of-word (CBOW) model and the skip-gram (SG) model (Mikolov et al., 2013)

CBOW

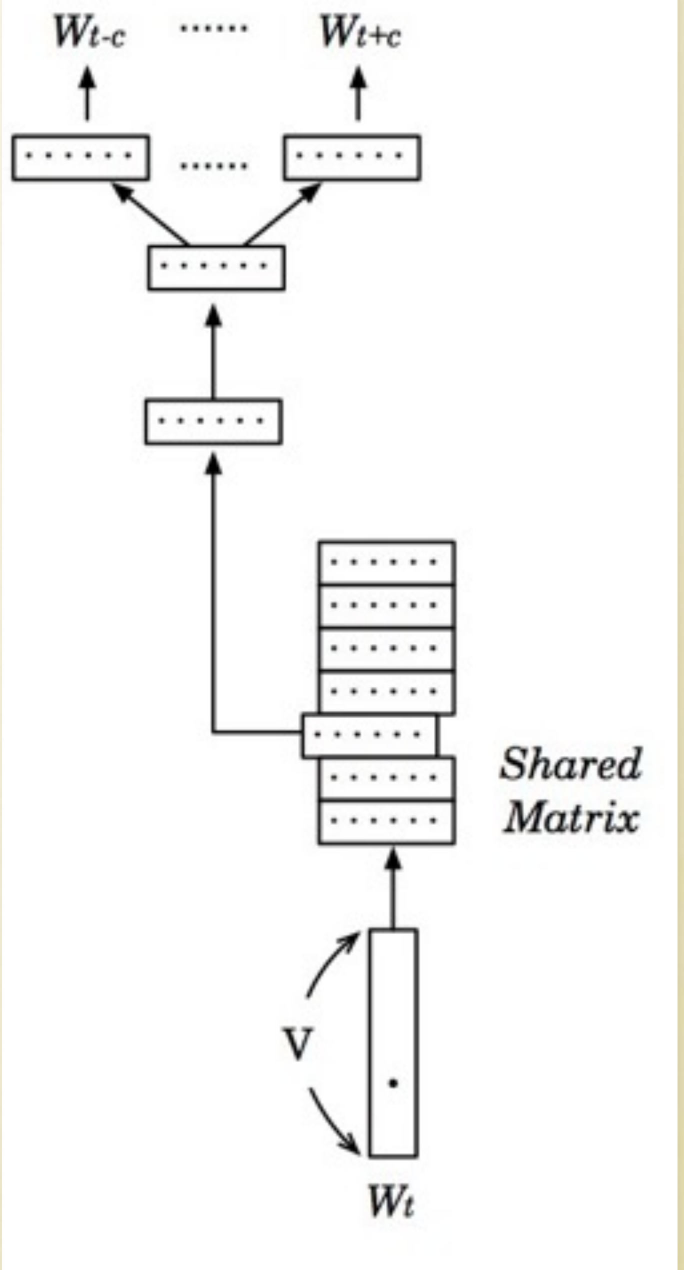
- Predict this word based on its neighbors
- Sum vectors of context words
- Linear activation function in hidden layer
- Output a vector
- Back-propagation to adjust the input vector and weights



(a) CBOW

Skip-gram (SG)

- Predict neighbors word based on this word
- Input vector of this word
- Linear activation function in hidden layer
- Output n other words
- Back-propagation to adjust the input vector and weights



(b) SG

From Word to Document

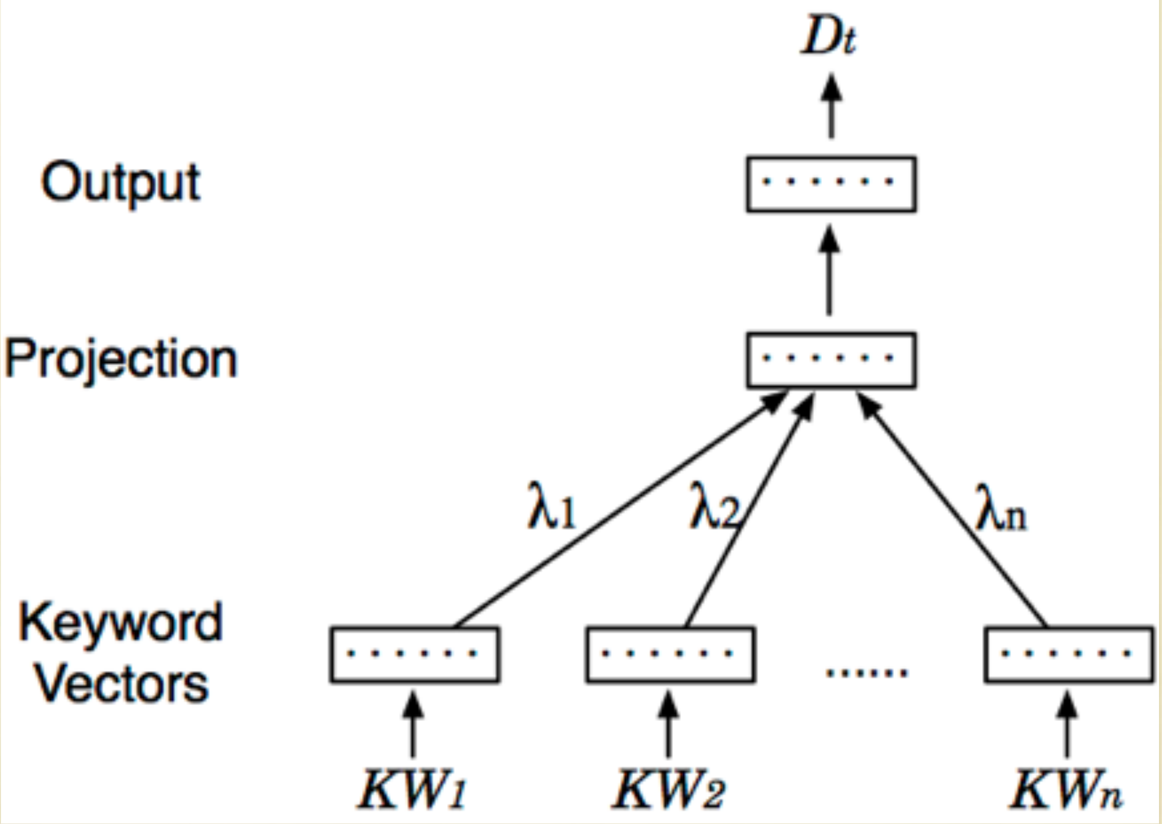
- By the same line of thought, we can represent a sentence/paragraph/document using a vector.

  (Le and Mikolov, 2014)

- A sentence or document ID is put into the vocabulary as a special word.
- Train the ID with the whole sentence/document as the context.
- *CBOW⇒DM, SG⇒DBOW*

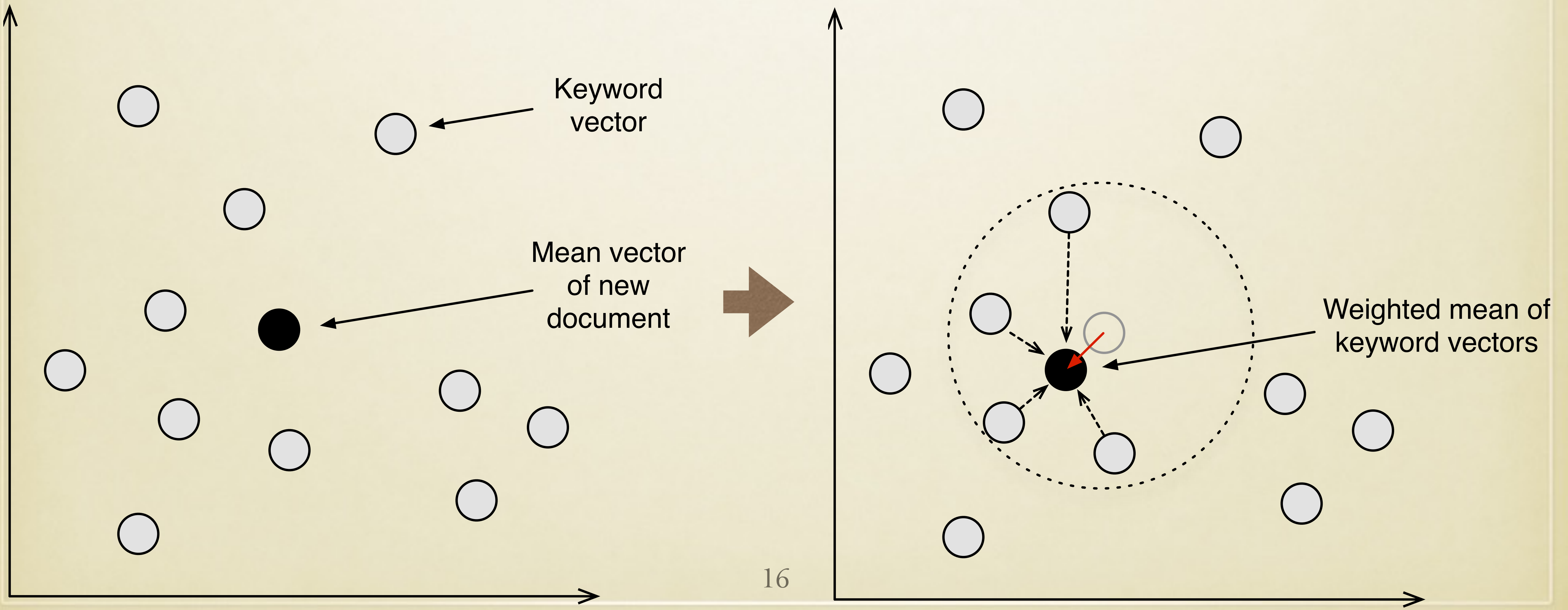
Novel Representation for Documents

- Distributed Keyword Vectors, **DKV**
- Rank keywords for each category using LLR
- A document is represented by the combination of keyword vectors
  - Weights of keywords are determined by LLR
- More discriminative



Unseen Documents

- An unseen document might contain no keywords
- We can represent it by using ***n*** nearest DKVs



Outline

- Introduction
- Previous Work
- Proposed Method
- Experiments
- Results & Discussion
- Conclusion

Corpus

- We collected a corpus of 100,000 Chinese news articles from Yahoo! online news
- Each article is categorized into five topics, namely, *Sports, Health, Politics, Travel, and Education*
- Training and testing sets both contain 50,000 documents, with equal amount of documents/topic

Experimental Settings

- DKV:
  - Train CBOW word vectors with 100 dimensions
  - Rank keywords using LLR
  - Weighted sum of keywords' vectors represents a documents for learning an SVM classifier
- Evaluation metric: F-1 score
- We test 1) against other classification methods, and 2) with various settings for the amount of keywords

Comparisons

- Naïve Bayes (**NB**)
- Vector space model (**VSM**)
- Latent Dirichlet allocation for representation with an SVM classifier (**LDA**)
- Two neural network-based representations (**DM** and **DBOW**) with the same dimensionality setting as DKV, and an SVM classifier
- Evaluation: F-1 score

Outline

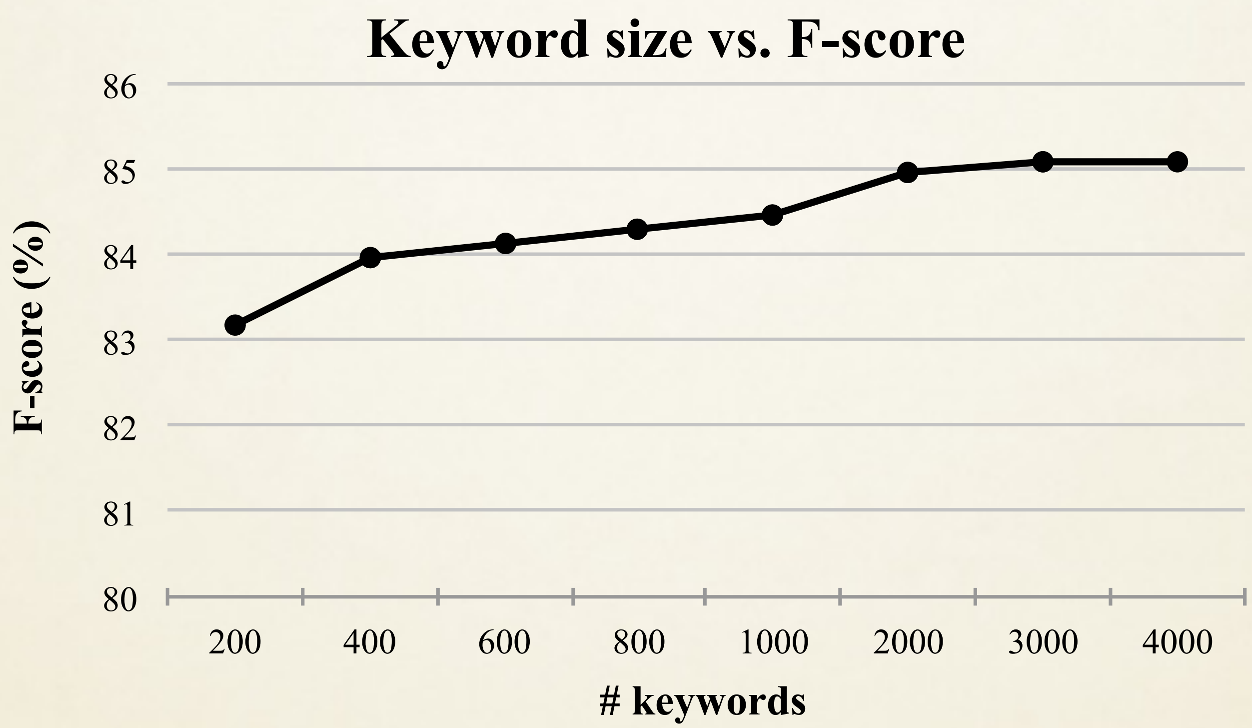
- Introduction
- Previous Work
- Proposed Method
- Experiments
- Results & Discussion
- Conclusion

Results I

- NB and VSM use only surface word weightings, thus fail to reach satisfactory performances
- LDA includes both local and long-distance word relations, leading to substaitial success
- Neural-network based methods have robust representation power
- DKV can successfully encode the relations between keywords and topics into a dense vector, leading to the best overall performance

Topic	NB	VSM	LDA	DM	DBOW	DKV
Sport	67.07	79.13	80.20	90.67	90.74	**92.22**
Health	40.41	63.65	80.35	86.73	86.67	**90.29**
Politics	42.86	66.89	67.31	85.41	85.70	**86.78**
Travel	42.52	66.31	**80.37**	74.08	74.40	72.01
Education	28.25	41.07	58.01	71.64	71.61	**74.54**
Average	44.22	63.41	73.25	81.71	81.82	**83.17**

Results II



- In the range from 200 to 4,000 keywords, F1-score is positively related to keyword size, however,
- The difference is not obvious (< 0.1%) when we reach a certain amount (~2,000 keywords)
- The contribution from keywords has saturated in our model, and simply adding more keywords would not lead to improvement

Outline

- Introduction
- Previous Work
- Proposed Method
- Experiments
- Results & Discussion
- Conclusion

Conclusions

- We present a novel model for text categorization using distributed keyword vectors as features
- Demonstrated the potential of strong representative power of neural networks and effectiveness of LLR in keyword selection
- More keywords do not equal to better performance, but maybe related to the nature of the corpus

Future Work

- Improve keyword selection method
- Deeper neural network for categorization
- Incorporate semantic information into word vectors
- Capture long-distance dependency
- Explore other applications for our method

Thank You

Questions or comments are welcomed!