Synergistic Union of Word2Vec and Lexicon for Domain Specific Semantic Similarity

Computer Science and Engineering Department
University of Moratuwa, Sri Lanka

University of London International Programmes
University of London

12th IEEE International Conference on

Industrial and Information Systems (ICIIS) 2017

December 15th-16th, 2017 | Peradeniya, Sri Lanka

OUTLINE

1. Introduction
2. Background Work
3. Proposed Methodology
4. Results
5. Conclusion and Future Work

INTRODUCTION

1. What are Word Embeddings
2. What is Word2vec
3. Lexical Semantic Similarity between Words

INTRODUCTION

- In word embedding, words or phrases from the vocabulary are mapped to vectors of real numbers.
- Word2Vec is a two layer neural network that produces a vector space model for a given piece of document
- Lexical Semantic similarity of two entities is a measure of the likeness of the semantic content of those entities
- This is calculated with the help of topological similarity existing within an ontology or lexicon such as WordNet

BACKGROUND WORK

1. Word Vector Embeddings
2. Word2Vec
3. Lexical Semantic Similarity Measures
4. Legal Information Systems

BACKGROUND

1. Word Vector Embedding using word to neighbouring word mapping

Pennington, Jeffrey and Socher, Richard and Manning, Christopher D, “Glove: Global Vectors for Word Representation,” EMNLP pp. 1532-1543, 2014.

2. A lexical database for topological similarity measures

Miller, George A and Beckwith, Richard and others, “Introduction to WordNet: An on-line lexical database,” International journal of lexicography pp 235--244, 1990.

3. Creating vector representations of words in the legal domain

Nay, John J, “Gov2vec: Learning distributed representations of institutions and their legal text,” pp. 2016.

Methodology

1. Text Lemmatization
2. Training Word2Vec Models
3. Lexical Semantic Similarity Enhancements
4. Neural Network Training
5. Experiments

METHODOLOGY

1. **Text Lemmatization**

- The linguistic process of mapping inflected forms of a word to the word's core lemma is called lemmatization
- Crawled all legal cases from FindLaw online repository
- Maintaining a separate vector for each inflected form of each word makes the model bloat up and consume memory unnecessarily
- For this task we use the Stanford CoreNLP library.

METHODOLOGY

2. Training Word2Vec Models

- Training of the word embeddings was the process of building a word2vec model.
- The text corpus consisted of 35000 legal cases collected from FindLaw.
- Stanford CoreNLP for preprocessing the text with tokenizing, sentence splitting, Part of Speech (PoS) tagging and lemmatizing.
- size (dimensionality) set to 200, context window size set to 10, learning model is CBOW, min-count is set to 5, training algorithm is hierarchical softmax.

METHODOLOGY

Generated Models

- Word2Vec(G) Model
- Word2Vec(LR) Model
- Word2Vec(LL) Model
- Word2Vec(LLS) Model

METHODOLOGY

Word2Vec(G) Model

- The model trained using the generic text corpus collected from the FindLaw website

Word2Vec(LR) Model

- The model trained using the pre-processed (without lemmatization) text corpus collected from the FindLaw website.

METHODOLOGY

Word2Vec(LL) Model

- The model trained using the pre-processed (with lemmatization) text corpus collected from the FindLaw website

Word2Vec(LLS) Model

- The neural network model trained using the pre-processed (with lemmatization) text corpus collected from the FindLaw website, which is enhanced with semantic similarity measures

METHODOLOGY

3. Semantic Similarity Enhancements

- Word2vec distances - taken from Word2Vec(LL)
- Wu and Palmer - similarity measures
- Jiang and Conrath - similarity measure
- Hirst and St-onge - similarity measure

METHODOLOGY

4. Neural Network Training

$$W = \{w_1, w_2, w_3, ..., w_n\}$$

$$D = \{d_1, d_2, d_3, ..., d_n\}$$

$$M = \begin{bmatrix} wup(k, w_1) & wup(k, w_2) & \dots & wup(k, w_n) \\ jcn(k, w_1) & jcn(k, w_2) & \dots & jcn(k, w_n) \\ hso(k, w_1) & hso(k, w_2) & \dots & hso(k, w_n) \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

$$V = \begin{bmatrix} D \\ SM \end{bmatrix}^T$$

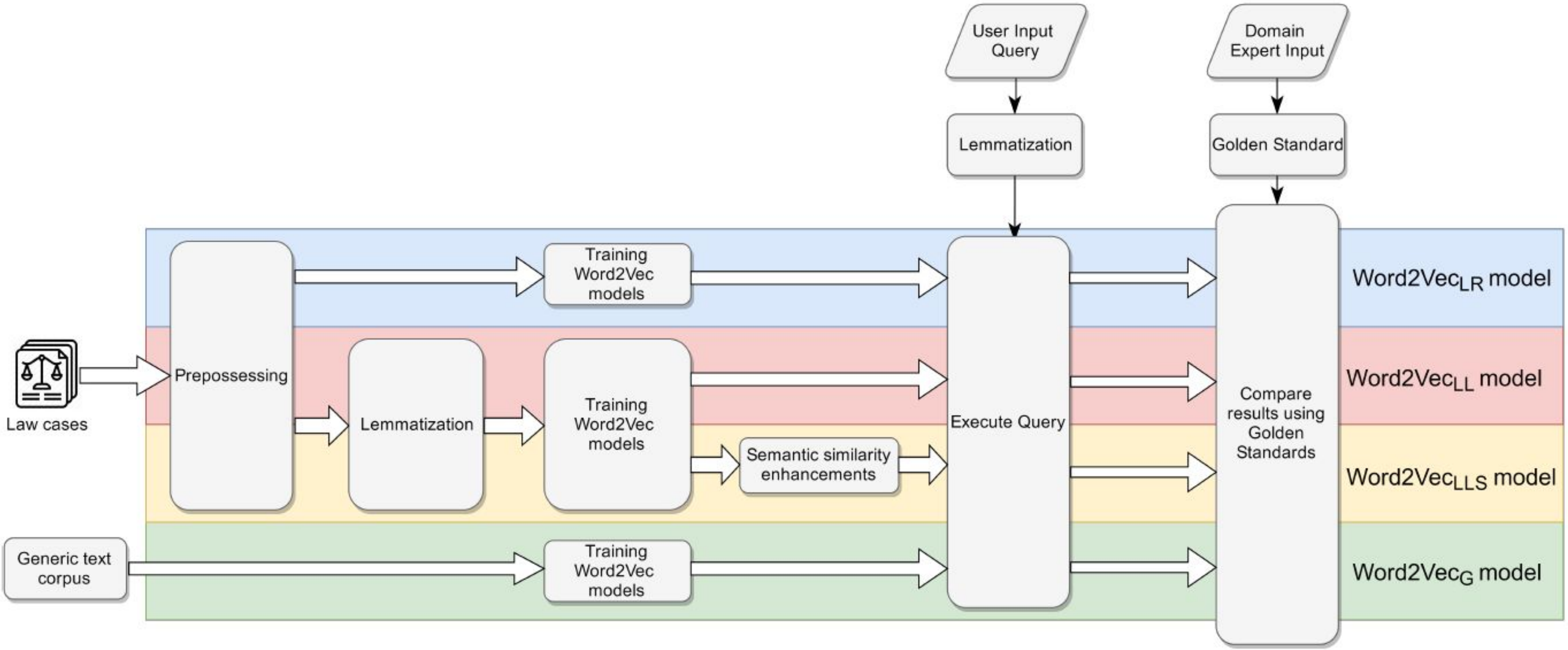
METHODOLOGY

5. Experiments

- Created a golden standard with the help of legal domain experts
- The accuracy levels of these experiments are measured in terms of precision and recall

METHODOLOGY

Flow of the complete Methodology



RESULTS

RESULTS

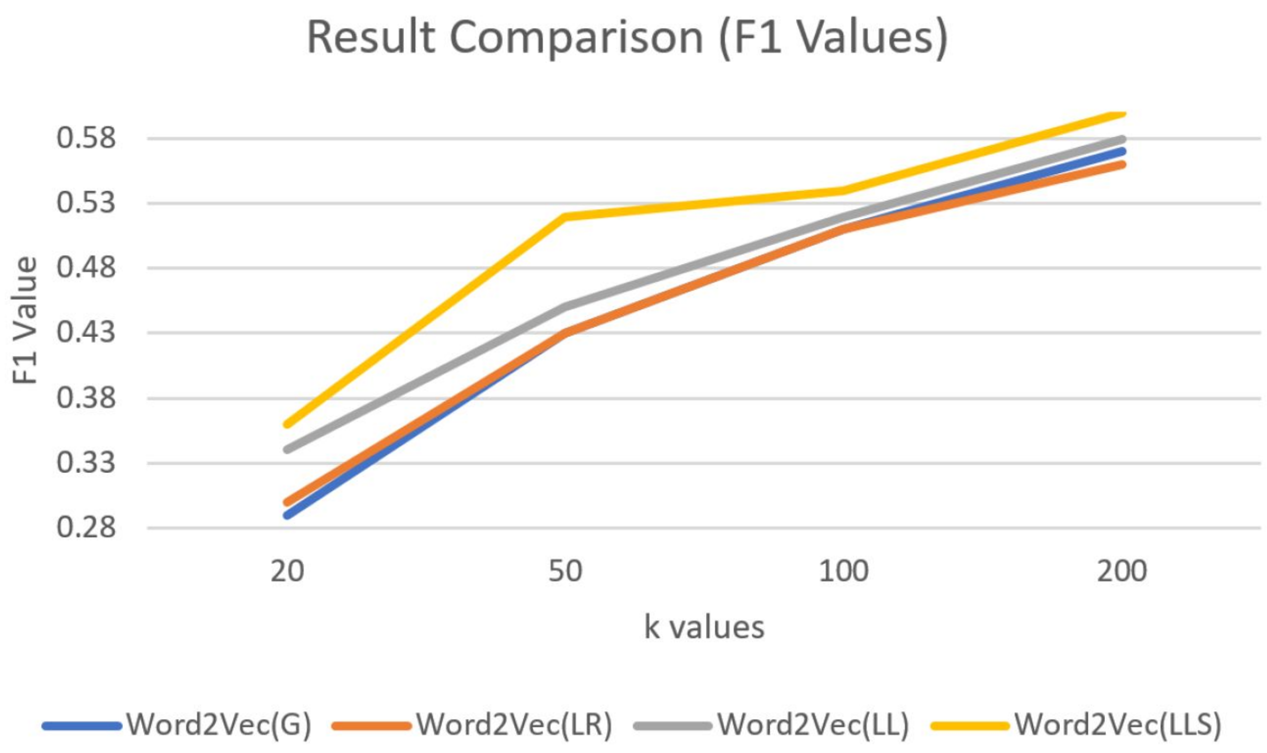
COMPARISON OF PERFORMANCE OF MODELS

RESULTS COMPARISON (P=PRECISION, R=RECALL)

Model	k=20			k=50			k=100			k=200		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
$word2vec_G$	0.57	0.19	0.29	0.62	**0.33**	0.43	0.67	0.41	0.51	0.74	0.46	0.57
$word2vec_{LR}$	**0.75**	0.19	0.30	0.71	0.31	0.43	0.74	0.38	0.51	**0.77**	0.44	0.56
$word2vec_{LL}$	0.73	0.22	0.34	0.72	0.32	0.45	**0.75**	0.40	0.52	0.76	0.47	0.58
$word2vec_{LLS}$	0.66	**0.24**	**0.36**	**0.73**	**0.33**	**0.52**	0.72	**0.43**	**0.54**	0.74	**0.50**	**0.60**

RESULTS

Comparison of precision, recall and F1 of the models



CONCLUSION & FUTURE WORK

CONCLUSION AND FUTURE WORK

- Domain specific document retrieval model, with enhanced vector representations and semantic similarity measures
- Ways to optimize semantic similarity measures using different lexicons of different languages
- Word vector embeddings on multi-lingual documents
- Domain Specific document vector embeddings

THANK YOU !!

Q&A