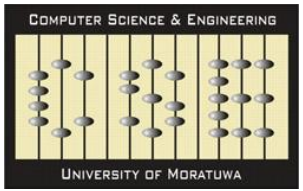
Deriving a Representative Vector for Ontology Classes with Instance Word Vector Embeddings



Computer Science and Engineering Department
University of Moratuwa, Sri Lanka

Seventh International Conference on Innovative Computing Technology (INTECH 2017)
August 16-18, 2017 | Luton, UK

OUTLINE

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

INTRODUCTION

1. Why a representative vector
2. Ontology and representative vectors

INTRODUCTION

- Selecting a representative vector for a set of vectors is a very common requirement in many algorithmic tasks.
- Traditionally, the mean or median vector is selected.
- An ontology is a “formal and explicit specification of a shared conceptualization”.
- Deriving representative vectors for ontology classes is an important problem in the domain of automatic ontology population and automatic ontology class labeling.

BACKGROUND WORK

1. Ontologies
2. Word Set Expansion
3. Word Embedding
4. Clustering
5. Support Vector Machines

BACKGROUND

1. Semi-supervised algorithm for concept ontology based word set expansion

N. H. N. D. de Silva, A. S. Perera, and M. K. D. T. Maldeniya, “Semi-supervised algorithm for concept ontology based word set expansion,” Advances in ICT for Emerging Regions (ICTer), 2013 International Conference on, pp. 125–131, 2013.

2. Distributed representations of words and phrases and their compositionality

T. Mikolov, I. Sutskever, K. Chen et al., “Distributed representations of words and phrases and their compositionality,” Advances in neural information processing
systems, pp. 3111–3119, 2013.

3. word2vec

https://code.google.com/p/word2vec/

Methodology

1. Ontology Creation
2. Training word Embeddings
3. Sub-Cluster Creation
4. Support Vector Calculation
5. Candidate Matrix Calculation
6. Optimization Goal
7. Machine Learning implementation for weight calculation

METHODOLOGY

1. **Ontology Creation**

- Legal ontology based on the consumer protection law.
- Findlaw as the reference.
- Manually added seed instances for all the classes in the ontology.
- Set expansion algorithm to expand the instance sets.

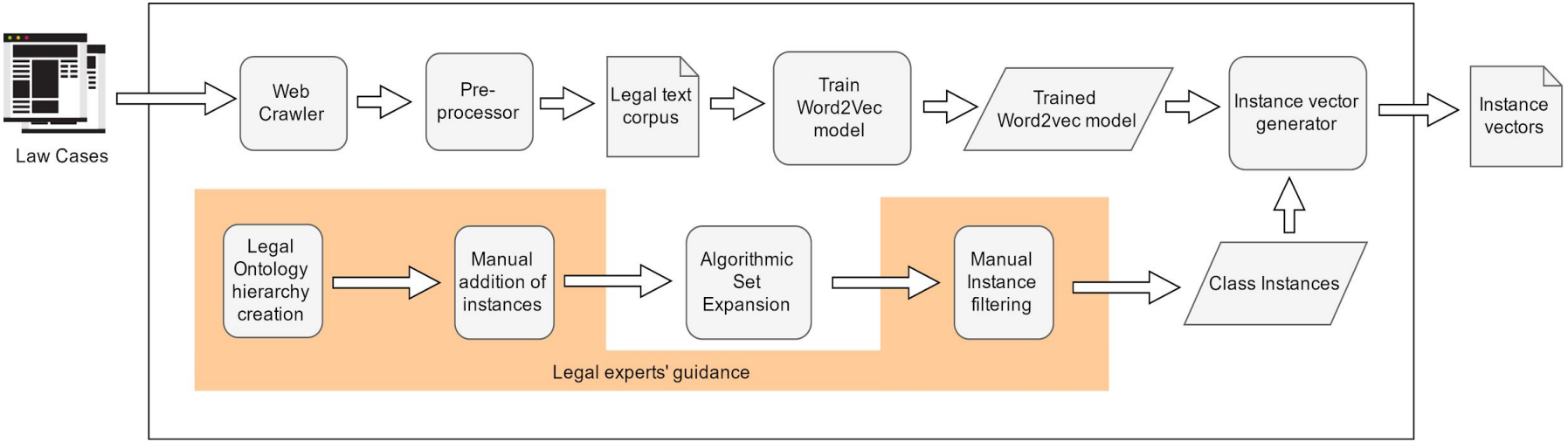
METHODOLOGY

2. Training word Embeddings

- Training of the word embeddings was the process of building a word2vec model.
- Stanford CoreNLP for preprocessing the text with tokenizing, sentence splitting, Part of Speech (PoS) tagging and lemmatizing.

METHODOLOGY

Flow diagram of the methodology for deriving instance vectors



METHODOLOGY

3. Sub-Cluster Creation

- By definition, the instances in a given class of an ontology is more semantically similar to each other than instances in other classes.
- But no matter how coherent a set of items is, as long as that set contains more than one element, it is possible to create non-empty subsets that are proper subsets of the original set.
- Which means we decided to divide the instances in a single class into two sub-clusters.
- K-means clustering with K=2.

METHODOLOGY

4. Support Vector Calculation

- The previous step yielded two subclusters with two unique labels.
- A SVM was given the individuals in the two sub-clusters as the two classes and was directed to discover the support vectors.

METHODOLOGY

5. Candidate Matrix Calculation

- Average support vector (C1)
- Average instance vector (C2)
- Class Median vector (C3)
- Sub-cluster average vectors (C4, C5)

* **Class name Vector (C0)** is obtained by performing word vectorization on the selected class's class name and it was used as our desired output.

METHODOLOGY

5. Candidate Matrix Calculation

- *Average support vector (C1)* - The average of the support vectors obtained in previous steps.
- *Average instance vector (C2)* - The average of the instance vectors of the relevant class.
- *Class Median vector (C3)* - Out of the instance vectors of a class, the median vector of them.
- *Sub-cluster average vectors (C4, C5)* - Averages of the instance vectors of sub clusters separately.

METHODOLOGY

6. Optimization Goal

- An optimal vector that represents the given class based on the optimization goal as follows:

$$Y = \frac{\sum_{i=1}^{M} C_i W_i}{\sum_{i=1}^{M} W_i}$$

Here, Y is the predicted class vector for the given class. M is the number of candidate vectors for a given class. Ci and Wi represents the ith candidate vector and the associated weight of that candidate vector respectively.

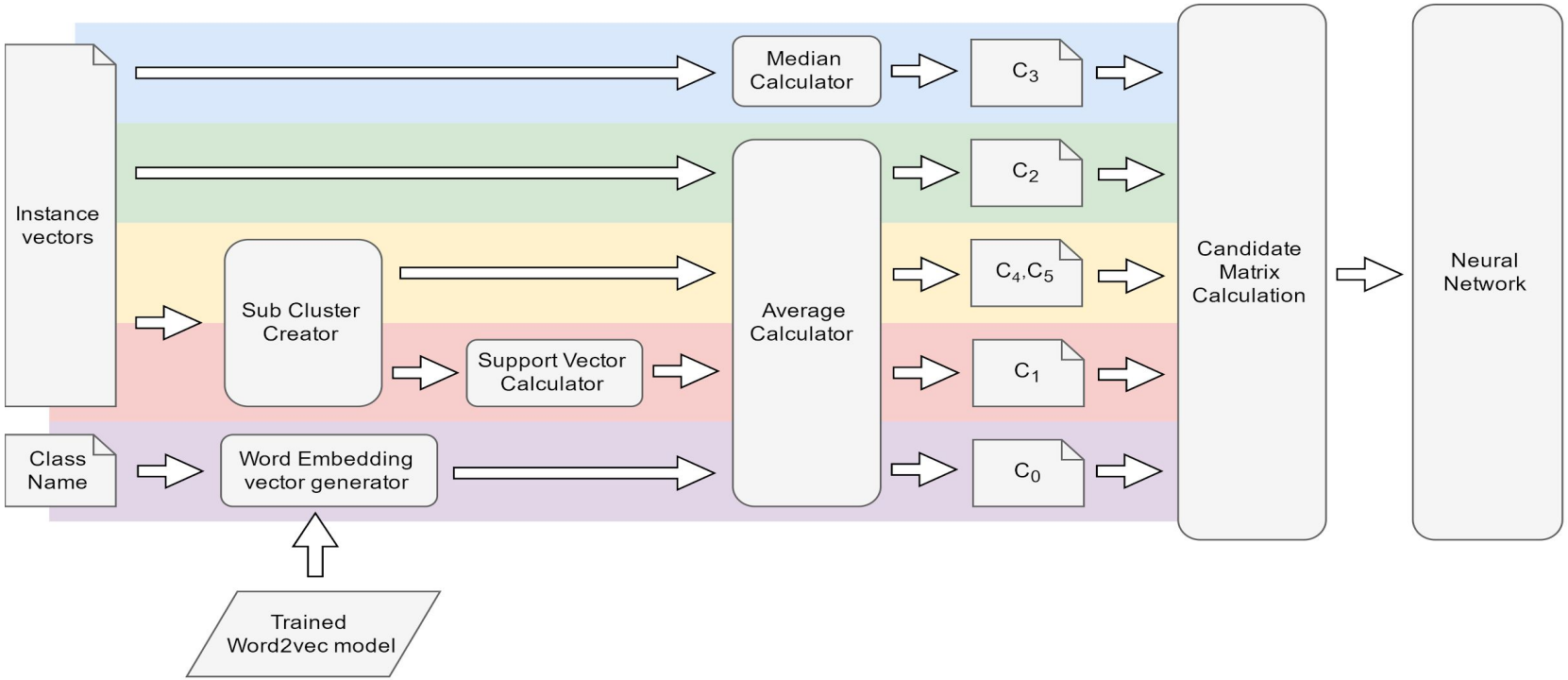
METHODOLOGY

7. Machine Learning implementation for weight calculation

- The main motive behind adding a weight for each candidate vector is to account for the significance of the candidate vector towards the optimized class vector.
- Machine learning to calculate the weight vector.
- The machine learning method used is a neural network.

METHODOLOGY

Flow diagram of the methodology for training and testing the neural network



RESULTS

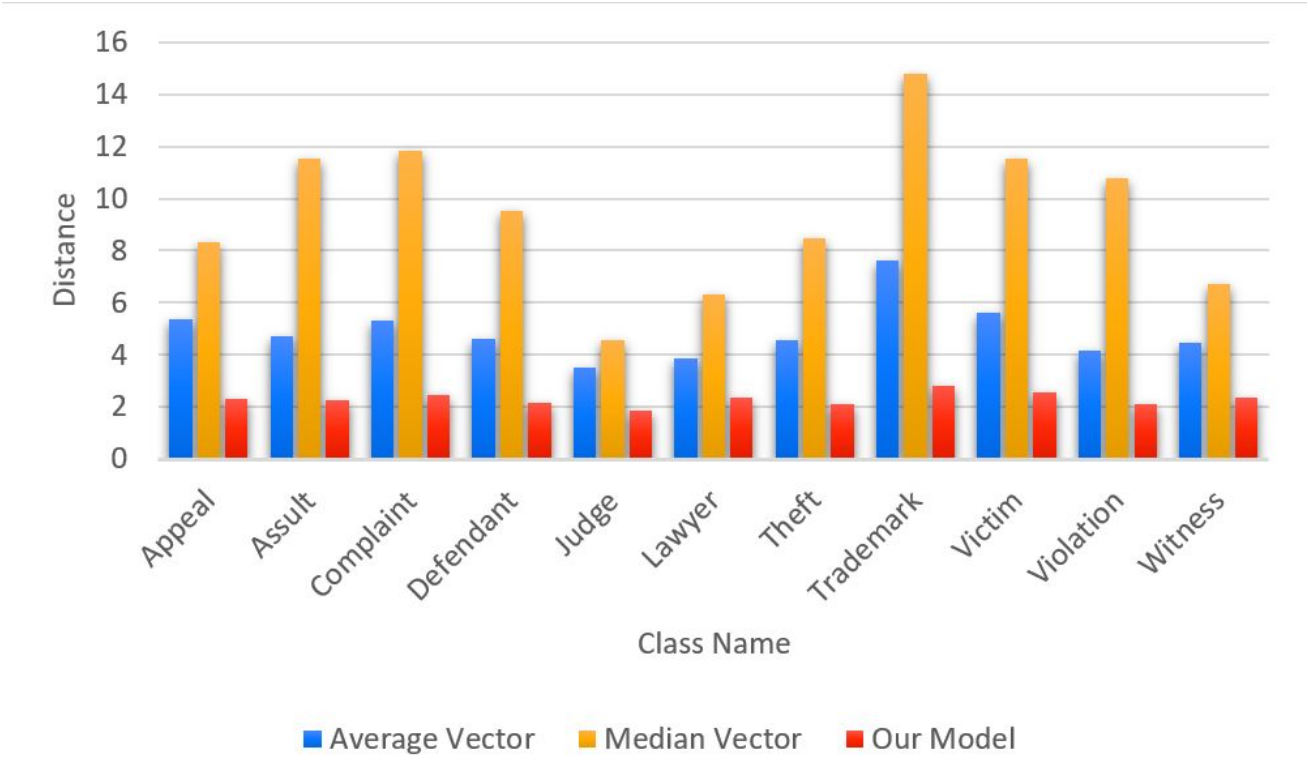
RESULTS

Distance measures of the classes and the average over 10 classes

	Average Vector	Median Vector	Our Model
Appeal	5.37	8.32	**2.31**
Assault	4.73	11.53	**2.27**
Complaint	5.33	11.82	**2.46**
Defendant	4.60	9.51	**2.15**
Judge	3.52	4.56	**1.84**
Lawyer	3.89	6.33	**2.35**
Theft	4.56	8.49	**2.13**
Trademark	7.63	14.79	**2.81**
Victim	5.64	11.52	**2.54**
Violation	4.18	10.80	**2.09**
Witness	4.46	6.73	**2.34**
Class mean	1.31	1.51	**0.82**

RESULTS

Comparison of the distance measures of representative vectors



CONCLUSION & FUTURE WORK

CONCLUSION AND FUTURE WORK

- This discovery will be helpful in mainly two important tasks in the ontology domain.
  - Further populating an already seeded ontology
  - Class labeling

- Word Set Expansion algorithm used has the weakness of being dependent on the WordNet lexicon.

THANK YOU !!