The University of Nottingham

PhD Programme

Automatic Idea Generation and Analysis Using NLP and ML Techniques

Author:
Haixia Liu

Supervisors:
Dr.Timothy Brailsford
Dr.James Goulding
Dr.Tomas Maul
Dr.Tessa Houghton

School of Computer Science. The University of Nottingham,
Malaysia Campus Jalan Broga, 43500 Semenyih
for the degree of PhD in Computer Science

Sponsored by

Crops for the Future Research Centre

January 2019

The University of Nottingham

Abstract

Faculty of Science
Computer Science

PhD of Computer Science

Automatic Idea Generation and Analysis Using NLP and ML Techniques

by Haixia Liu

Abstract

Ideas are the fundamental way in which information is conveyed in written text. This research investigates the discovery and extraction of ideas from corpuses of scientific literature. There are several elements to this work: (1) the functional definition of ideas; (2) the computation of novel ideas; (3) the representation of ideas; (4) the construction of a ground truth dataset; and (5) the use of citations as an idea container.

Ideas are defined as a <problem, solution> pair, where the problem and solution are represented by noun phrases, or a sequence of words. As a result of this, the task of idea detection is broken down to problem and solution extraction. The task of idea extraction is similar to Named Entity Recognition (NER), where the problems and solutions may be seen as special entities. These techniques worked well although the results contained a lot of noise that need to be removed.

Automatic idea generation was conducted using a dataset from the Journal of *Science*. Old ideas were defined as the existing <problem, solution> pairs in the same abstract and new ideas were generated by predicting new links between problems and solutions that do not occur together in one abstract. Evaluation was performed using metrics that are widely used in information retrieval. The F1 scores (higher than 0.90) provides good evidence that the proposed method is capable of generating useful ideas.

A ground truth data set that contained <problem, solution> pairs was constructed from the publications of the International Conference on Neural Information Processing Systems and the Journal of Machine Learning Research. This data was annotated by human volunteers, and it was used for training idea detection models using Conditional Random Field (CRF) and Long-short Term Memory (LSTM). To evaluate the performance of the models, the precision and recall were computed.

Idea analysis was studied by analyzing citations, which are considered to be containers for ideas. Word vectors were used to represent the citations for the purpose of classifying citation sentiment, and a method was developed to measure the sequence of citation sentiment. This method for analyzing internal citation sentiment sequence worked well (with F1 measure 0.86).

Acknowledgements

Many thanks to my supervisors for giving me this interesting research topic to work with. Appreciations for their guidance and advices on how to conduct this research. This thesis would not have been completed without the supports of the following parties:

Thanks for Crop For the Future (CFF) research center [1] for funding this project.

Thanks for the School of Computer Science in the University of Nottingham Malaysia Campus (UNMC) and Jubilee Campus. Thanks for the Horizon Digital Economy Institute.The seminars and scientific talks are very helpful.

Thanks for the Graduate School of UNMC. The professional training courses provided by them are very helpful. The coffee-tea parties they organized are the friendly social events on campus.

Thanks for the Ubiquitous Knowledge Processing Lab (UKP Lab) at the Computer Science Department of the Technische Universitt Darmstadt, where I learned deep learning techniques.

Thanks for my schoolmates for being great accompanies on my journey for pursuing this PhD. The conversations between me and the lab members always inspired me and encouraged me.

Thanks for the accommodation office of UNMC to offer me the Hall Tutor job, which enabled me to gain the skills as a leader.

Thanks for the encouragements from my aunt, my cousins, my childhood friends.

[1] http://www.nottingham.edu.my/CFFRC/index.aspx

Contents

Acknowledgements iii

List of Tables vi

List of Figures viii

1 Introduction 1
- 1.1 Background of the study . . . 1
- 1.2 Research problem . . . 9
- 1.3 Research objectives . . . 9
  - 1.3.1 The research goals. . . . 9
  - 1.3.2 Contributions. . . . 12
- 1.4 Research scope . . . 13
- 1.5 Structure of the rest of the thesis . . . 13

2 Literature Review 16
- 2.1 Idea mining . . . 16
  - 2.1.1 Data mining . . . 17
  - 2.1.2 Machine Learning . . . 22
  - 2.1.3 The fundamental techniques of NLP . . . 30
  - 2.1.4 Some NLP applications . . . 35
- 2.2 Novel idea computation . . . 36
  - 2.2.1 Computational Creativity . . . 37
  - 2.2.2 Information retrieval . . . 40
- 2.3 Idea flow analysis . . . 44
  - 2.3.1 Word2vec . . . 44
  - 2.3.2 Argumentative zoning . . . 45
  - 2.3.3 Sentiment analysis . . . 48
  - 2.3.4 Texture analysis in image processing . . . 50
  - 2.3.5 Sentiment Level Co-occurrence Matrix (SLCM) . . . 51
  - 2.3.6 Eddy Dissipation Rate . . . 52

3 Research Methodology 54
- 3.1 Dataset review . . . 54
  - 3.1.1 Dataset for idea mining and novel idea computation . . . 54

3.1.2 Dataset for detecting ideas using neural networks 55
3.1.3 Dataset for Argumentative Zoning (AZ) 57
3.1.4 Dataset for citation sentiment analysis 59
3.1.5 Dataset for idea flow analysis 60
3.2 Methodology 62
3.2.1 Idea mining and novel idea computation 62
3.2.2 Idea detection using neural networks 73
3.2.3 Argumentative Zoning 80
3.2.4 Citation sentiment analysis 86
3.2.5 Idea flow analysis 87

4 Results and Discussion 101
4.1 Idea mining and novel idea computation 101
4.1.1 Result 101
4.1.2 Discussion 103
4.2 Idea detection using neural networks 105
4.2.1 Result 105
4.2.2 Discussion 108
4.3 Argumentative Zoning 109
4.3.1 Result 109
4.3.2 Discussion 111
4.4 Citation sentiment analysis 112
4.4.1 Result 112
4.4.2 Discussion 113
4.5 Idea flow analysis 114
4.5.1 Result 114
4.5.2 Discussion 117

5 Conclusion and Future work 120
5.1 Idea mining and novel idea computation 120
5.1.1 Conclusion 120
5.1.2 Future work 121
5.2 Idea detection using neural networks 122
5.2.1 Conclusion 122
5.2.2 Future work 122
5.3 Argumentative Zoning 124
5.3.1 Conclusion 124
5.3.2 Future work 124
5.4 Citation sentiment analysis 125
5.4.1 Conclusion 125
5.4.2 Future work 125
5.5 Idea flow analysis 126
5.5.1 Conclusion 126
5.5.2 Future work 127

List of Tables

1.1 Observations of the sentences containing the word *idea* 4
1.2 Examples of how new discoveries change the knowledge pool 8
1.3 Examples of how novel ideas change the knowledge pool 8
1.4 The expected and the actual outcomes of the research goal: idea detection from abstracts. 11

2.1 Examples of annotated sentences with their rhetorical status 48
2.2 Examples of positive and negative citations. 49

3.1 Examples of title and abstract from journal Science. 55
3.2 Examples of ACL Anthology and AZ Corpus. 58
3.3 Characteristics of word embeddings based on different model and dataset 58
3.4 Examples of $dataset - basic$ and $dataset - implicit$. 60
3.5 Samples of $Data_I$ 61
3.6 Samples of citation sequence with raw citation counts 61
3.7 Samples of $Dataset_G$ 62
3.8 Noun-Phrase filtering terms 67
3.9 S-phrase cue terms. The term "method" was used as a seed, with the set of trigger words being expanded through a combination of synonym extraction via *www.thesaurus.com* and isolating nearest neighbours using *Word2vec* (see https://code.google.com/p/word2vec/). 68
3.10 Examples of known-idea triplets from a sample corpus of 20 documents: 71
3.11 Different settings of the hidden layer nodes resulted in different outcomes. 76
3.12 Distribution of rhetorical categories 85
3.13 Observations of citation sentiment changes - Internal 90
3.14 Examples of citation sentiment sequences generated from one paper. 91
3.15 Examples of citation sentiment sequences (shown in vertical) generated from a collection of papers that cited the same paper. $G-1$ to $G-2$ in equation 3.31 and 3.32 represent the methods for computing the overall sentiment score described in section 3.2.5 93

4.1 Examples of novel idea-pairs generated from journal Science. 102
4.2 OCC performance (precision/recall/F-measure) 102

4.3 CON performance (precision/recall/F-measure) 102
4.4 IF-IDF performance (precision/recall/F-measure) 102
4.5 Results generated by the model CWE-NN with respect to different training Epochs using pretrained word vectors on ACL300. 105
4.6 Results generated by the model CWE-NN with respect to different training Epochs using pretrained word vectors on GoogleNews. . . 106
4.7 Results generated by the model Bi-LSTM-CRFs. ACL300 represents using pretrained word vectors based on ACL300; GoogleNews represents using pretrained word vectors based on GoogleNews; Non-pretrained means the embeddings are not coming from pretrained word2ved model. 106
4.8 The annotations generated by the Bi-LSTM-CRFs model for the abstract in the paper published by Yang et al. (Yang et al., 2016) . 107
4.9 Performances of sentence classification based on different models per category with feature size 100 on the corpus ACL+AZ (precision/recall/F-measure) 109
4.10 Performances of sentence classification based on different models per category with feature size 100 on the corpus ACL+AZ (precision/recall/F-measure) - continue Table 4.9 109
4.11 Performance of sentence classification for the model $AVGWVEC$ based on different text corpus per category with feature size 100 (precision/recall/F-measure) 109
4.12 Performance of sentence classification for the model $AVGWVEC$ based on different text corpus per category with feature size 100 (precision/recall/F-measure) - continue Table 4.11 110
4.13 Performance of sentence classification with the model $AVGWVEC$ based on different feature sizes on the corpus ACL+AZ (precision/recall/F-measure) 110
4.14 Performance of sentence classification with the model $AVGWVEC$ based on different feature sizes on the corpus ACL+AZ (precision/recall/F-measure) - continue Table 4.13 110
4.15 Performance of citation sentiment classification on dataset-basic. . . 112
4.16 Performance of implicit citation sentiment classification. 113
4.17 Performance of classifying positive and negative citations. 113
4.18 Results (F1 scores) of ICSS classification using SVM. 115
4.19 Results of GCSSs clustering using Kmeans (using the seven metrics derived by EDR). Each cell contains the cited paper ID and the cluster label generated by edr_{value} ($pratio_l$ / $countcitations_l$ /g_{value}), where $pratio_l$ is demonstrated in the equation 3.31, which is the fraction of positive citations among all the citations. $countcitations_l$ is explained in the equation 3.30. It is the difference between the number of positive citations and the negative ones. g_{value} considers sentiment scores that are accumulated from each citing papers. The 20 cells in this table are organized according to the sequences position in Figure 4.1. 117

List of Figures

1.1 Where is the position for Idea in the DIKW pyramid? 7
1.2 Road map. 15

2.1 Business driven data process (M. Brown, 2004). It indicates that different data sources can be structured into different data format. Different data formats may or may not require different data mining techniques 19
2.2 Clustering stages (Rosenberg et al., 2005). 25
2.3 Schematic representation of the batch training approach with Expectation-Maximization (EM) (Rosenberg et al., 2005). 27
2.4 Schematic representation of the batch training approach with incremental self-training (Rosenberg et al., 2005). 28
2.5 Feature prototypes for detecting sentiment change. All of the prototypes contain two units, representing the adjacent two citations. The white-white (black-black) pattern stands for there is no sentiment change and both of the adjacent citations are positive (negative); the white-black pattern represents the sentiment change from positive to negative and the black-white pattern means the sentiment changes from negative to positive. 53

3.1 An example of human annotated abstract. The *problem* (*solution*) sequences were highlighted in yellow (blue). The sequence underlined is the solution candidate that was discarded by the annotator. 57
3.2 System flow - each of the method's processing steps are listed on the right hand side, with corresponding stages in real-life brainstorming sessions mapped on the left. 66
3.3 The *problem* (*solution*) sequences were highlighted in yellow (blue) . 74
3.4 The architecture of the CWE-NN model, which is typical artificial neural networks. 76
3.5 The architecture of the Bi-LSTM-CRF model. l_i (r_i) represents the word i and its left (right) context vector. The two vectors are concatenated forming the representation of the word i in the context c_i. The top layer (B-P, I-P, I-P, I-P) is the CRF layer. The arrows indicate the information (originally from the input text and carried out indirectly by matrix to the top layer) flow. 80
3.6 Using white-black prototype to analyze sequence [1, -1, 1, 1, -1]. . 96

3.7 Sentiment level co-occurrence matrix for the sequence [1, -1, 1, 1, -1]. The pair of cells in a row reflect the sentiment change: the [1, 1] and [-1, -1] pattern stands for there is no sentiment change and both of the adjacent citations are positive or negative; the [1, -1] pattern represents the sentiment change from positive to negative and the [-1, 1] pattern means the sentiment changes from negative to positive. 97

3.8 Normalized sentiment level co-occurrence matrix for the sequence [1, -1, 1, 1, -1]. 97

3.9 Plots of GCSSs (The cited-paper IDs are : J93-1007 (top-left), J90-1003 (top-right), W05-0909 (bottom-left) and W04-1013 (bottom-right). where the horizontal-axis represents the length of the sequence and vertical-axis is the normalized sentiment value $pratio_l$ (unnormalized ones are computed by Equation 3.30). The range of the value is 0-1, where 1 indicates strong positive and 0 indicates strong negative. The place without bar means the The $pratio_l$ value is 0. Clearly, the bottom two sequence are subjectively controversial. Four sequences with detailed $pratio_l$ values are shown in the Table 3.7. 98

4.1 Plots of GCSSs, where the horizontal-axis represents the citations overtime (with respect to a specific cited paper.) and vertical-axis is the normalized sentiment value $pratio_l$ (The range of the value is 0-1, where 1 indicates strong positive and 0 strong negative). The label of each graph could be found on Dr.Athar's corpus page and the details of the citations information to construct the graph is demonstrated on the html file accordingly. For example: in the top-left graph, the cited paper with the label *p07-1033* got 6 citations, which are ordered by the citation published time. The bottom-right citation *p04-1041* has 4 citations. The citation normalized sentiment value is a single point according to the vertical-axis, but for visualizing the changed, the points were connected by lines, which form the curves on the graphs. 115

Chapter 1

Introduction

1.1 Background of the study

Ideas are fundamental to all textual information, to the extent that conveying ideas is the reason why we write. It is therefore an interesting and important question to ask whether or not it is possible to use computers to extract and analyse ideas. Ideas are aspects of human thoughts that are independent of words or languages. There are, nonetheless, patterns in text that represent ideas. People can recognise these patterns and we call that reading. If people can represent patterns in data then maybe machines can too. The importance of idea mining leads to the actions in this work: examining techniques that might be used to enable machines to recognise patterns that we call ideas.

Motivation. The process of attacking problems by first canvassing participants for spontaneous ideas, collating their responses and distilling the results, is often referred to as *brainstorming*. The term, as popularized by Osborn (Osborn, 1953) and expanded upon by Kling (Kling, 1990) and Jessop (Jessop, 2002), now corresponds to a well-known set of guidelines for generating creative solutions that entail: discussion of the problem; unconstrained consideration as to how best to solve the problem; screening of the contributions; and, finally, commitment to action. While this approach to problem solving has traditionally required active

human participation, in this work the following challenge is explored: *given the inordinate amount of scientific literature now accessible via the web, is it possible to automate the brainstorming process via machine learning?*

While the idea of supporting the ideation process via technology is not new (the term *Computer-Assisted Brainstorming* was coined three decades ago (Hollander, 1984)), prior research has focussed on visualization tools, organizational applications and associated Human-Computer Interaction challenges (von Hardenberg & Bérard, 2001; Biemann, Böhm, Heyer, & Melz, 2004, 2006). However, text mining and computational linguistic techniques have now progressed to the point that notions of automatically extracting information from text and recognizing the links between underlying topics and concepts has become commonplace (Tan et al., 1999; Allan et al., 1998; Ding & Chen, 2014). This brings with it opportunity to not only provide support tools for ideation process, but to actually generate *ideas* themselves.

Generating novel ideas from the automated processing of mass corpora of scientific text requires us to address several conceptual problems. The first issue is that the term *'idea'* itself is not at all well-defined from a comprehension perspective (Hare & Milligan, 1984; Jitendra et al., 1998). Second, new ideas are built upon domain knowledge that is extremely hard, if not impossible, to formalize (Thorleuchter & Van den Poel, 2012). Third, ideas from different domains exhibit widely varying characteristics; and finally, commonly used methods for ideation, such as the Gordon technique and expertness (Taylor, 1961) are very difficult to computerize [1]. These issues imply that obtaining perfect solutions to problems without human input is unrealistic. However, there is much potential in addressing the sub-task of generating *idea candidates*. Using a functional definition of an "idea" as a <*problem*,*solution*> pair (in the vein of (Thorleuchter & Van den Poel, 2012)), an algorithmic approach to idea formulation was presented. This method breaks the task at hand into the following components: 1. a stage of text mining and linguistic processing of mass scientific corpora; 2. a supervised classification

[1] Gordon technique involves a process of progressively more detailed revelation, to avoid defining the problem too soon and limiting possible solutions.

stage to isolate problem and solution concepts; and 3. a stage of re-combination via collaborative filtering, which outputs novel idea pairs for researchers to consider. This approach is evaluated using a subset of publications from the journal *Science*. With a corpus of papers that cut across multiple disciplines, it is hoped that some of the idea candidates produced by the system will assist with the sort of cross-disciplinary ideation that is difficult to generate by conventional means.

Idea detection and novel idea creation are new and interesting fields in the natural language processing (NLP) and computational creativity research community. Idea detection is a process of extracting useful information from different modalities. Idea creation is a task of generating novel and interesting ideas based on existing linguistic structures. Both idea detection and idea creation are challenging due to the reasons: (1)It is difficult to define the *idea* in a concrete form; (2) The nature of an idea takes different forms; (3) A substantial of ground truth datasets that contain ideas are needed and yet they are not available; (4) The novelty of an idea is not easily to be evaluated.

The following contents of this chapter describe the idea definition, datasets containing ideas, the usefulness and novelty of an idea, the relation between idea and knowledge. The contributions and the structure of the work will be presented.

The definition of an idea. The Oxford English Dictionary defines idea as an abstract or eternally existing pattern or archetype of any kind of thing.

In philosophy, ideas are either to be mental representational images of some object or abstract concepts. It is a specific thought or concept that arises in the mind of a person as a result of thinking[2].

In technology (Thorleuchter, 2008) and social behavior domains (Thorleuchter et al., 2011), idea is defined as the pattern of means and ends occur together. This work adopts the definition of an idea as the pair of <means, ends>. Formally, the idea is defined as an <problem, solution> pair.

[2] *https://en.wikipedia.org/wiki/Idea*

TABLE 1.1: Observations of the sentences containing the word *idea*

Source	Example
wiki	The idea of animism was developed by the anthropologist Sir Edward Tylor in his 1871 book Primitive Culture,[1] in which he defined it as "the general doctrine of souls and other spiritual beings in general." According to Tylor, animism often includes "an idea of pervading life and will in nature"
wiki	Before understanding Aristotle's views on dreams, first his idea of sleep must be examined
publication	The key idea of the new design style is to limit both the upper and lower bounds of the voltage swing at the internal dynamic node
publication	The main idea is to accept those sequences appearing frequently and possessing high co-occurrences with consequents as indicative ones

The patterns of ideas. Unlike object detection from images (e.g.: face recognition), where the targeted objects are well shaped, the research of idea detection from text is not easily conducted. The pattern of an idea may vary, depending on the context. The pattern of a social idea and the pattern of a technical idea may take different forms. In order to discover the patterns of ideas, two sets of sentences that contain the word *idea* explicitly were collected, one of which is from wikipedia and the other one is from scientific publications. Examples of idea sentences are shown in the Table 1.1. The first two examples demonstrated some of the idea forms that emphasize will and concepts. In comparison, the two cases shown in the bottom of the Table 1.1 indicate that there are ideas developed for certain purposes. This research studies the idea pattern that embraces a purpose, which aims at solving certain problems, more specifically, extracting problem and solution information from scientific publications.

The Idea-datasets containing ground truth. Idea detection with computer algorithms is a young research field (only since 2010 (Thorleuchter, Van den Poel, & Prinzie, 2010b)) according to my knowledge, thus, there are not enough ground truth datasets for the experiments. The most relevant dataset to this work is the one in the paper (Christensen et al., 2017), where they announced 137 idea texts. Their dataset doesn't have problem and solution information in each piece

of text. Two sets of text corpus from publication abstracts will be constructed: (1) the mixed-topic dataset obtained from the *www.webofknowledge.com*; (2) the machine learning related dataset obtained from the Neural Information Processing Systems conference (NIPS). The reason to choose mixed-topic dataset is to generate novel ideas from multiple domains. The reason to choose NIPS as the data source for problem and solution detection is that NIPS holds technological papers, which are expected to contain many problems and solutions.

The first collected datasets are used for two distinct purposes: noun phrases extraction and novel links prediction between the noun phrases. The purpose of the second dataset is for collecting human annotations. Annotators are all researchers and most of them are from computer science, others are from Department of Geography. They highlighted the problem and solution pairs from the abstracts. The annotated NIPS abstracts are used for training a machine learning mode that can detect problem and solution automatically.

The usefulness and novelty of an idea. In paper (Connolly et al., 1993), they defined a novel idea as one that is rare, unusual, or uncommon. The generated new problem and solution pairs are not necessarily novel. In order to evaluate the new <problem, solution> pairs from the usefulness and novelty perspectives, human trials are usually adopted. But due to the subjective opinions introduced to the evaluation system, it is difficult to capture the unique usefulness and novelty patterns. This work is based on an assumption that the accepted publications contain useful and novel ideas, thus, it is possible to evaluate the computed ideas using the metrics that are used in information retrieval (IR) in such a way that the entire dataset is split into two parts, one of which is used as the information resources to generate new ideas and the other part is used for evaluation. If the generated ideas occur in the later part, they would be considered as useful and novel. The disadvantage of this method is, it is difficult to frame all the resources for both of idea generation and evaluation. As a result, the perceived useful and novel ideas maybe already exist in other resources and thus misclassification occurred. This is the limitation of this work, which can generated relatively novel

ideas within the corpus that's collected in this work. Also, the usefulness of the generated ideas needs further investigation by researchers.

The Hierarchy of Knowledge. In order to utilize rules derived from nature, it is important to study and organize the observations. The term knowledge management was proposed by American educator Nicholas L. Henry (Henry, 1974), who also distinguished among data, information, and knowledge. Although the origin of the Data, Information, Knowledge, Wisdom (DIKW) pyramid is uncertain, Milan Zeleny and Russell Ackoff have been credited with proposing the DIKW pyramid concept. The idea introduced in this work is composed of two elements: problem and solution. The relation between the problem and solution could be $solve$ and $possibably - solve$, where the former one is the guide for mining ideas and the later one is the motivation for generating novel ideas in this work. Since the compositions of the problems and solutions are concepts that are derived from data, it would be useful to understand the tier interposed between the components in the DIKW pyramid and idea. Where does the idea fit in the DIKW pyramid (illustrated in the Figure 1.1) is an interesting question for understanding the impact of ideas.

(1)Data. Data can be defined as signs that are representing stimuli received by some modalities, such as human and computer sensing systems. The elements of the hierarchy to knowledge forms are know-nothing, know-what, know-how, and know-why (Zeleny, 1987), thus, how to make sense of data becomes more and more important. In this work only the data from scientific publications is considered.

(2)Information. Information is related to data and knowledge, but information is intended to answer questions such as who, what, where, how and when (Ackoff, 1989). This work focuses on structuring the unstructured data from the publications for answering the questions *what and how*. The motivation of asking the question *what* is to discover the concepts representing the problems and solutions. The purpose of answering the question *how* is to find the solutions for solving

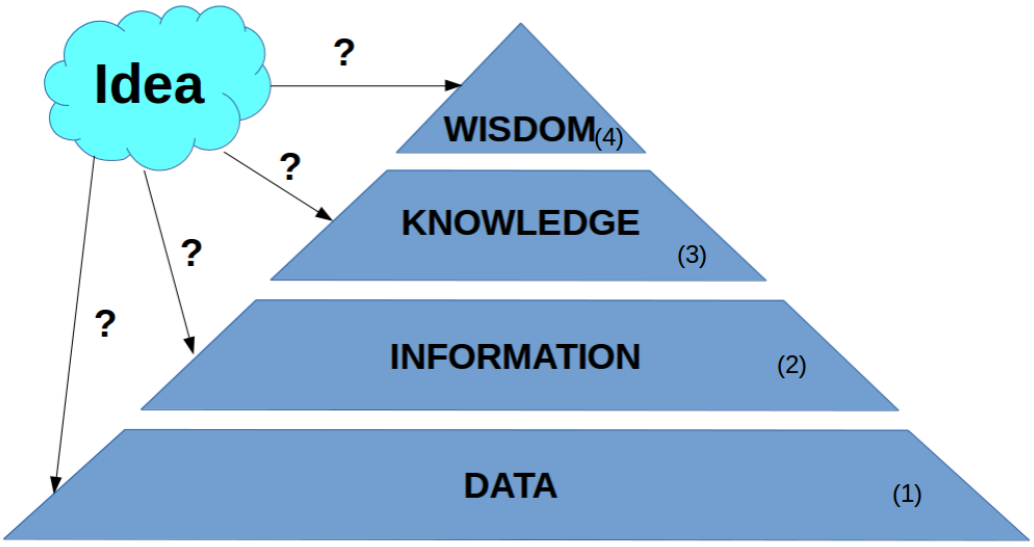


FIGURE 1.1: Where is the position for Idea in the DIKW pyramid?

certain problems, specifically, the question is: how did the author(s) solve the research problem in the published paper? and how to solve the old problem with a novel solution? Note that *novel* indicates either the problem or solution is novel.

(3)Knowledge. Knowledge can be perceived as processed information in such a way that the information is structured, organized and synthesized. One of the popular definition of knowledge reflects the variations of how people view knowledge. It says that knowledge is a fluid mix of framed experience, values, contextual information, expert insight and grounded intuition that provides an environment and framework for evaluating and incorporating new experiences and information. It originates and is applied in the minds of knowers. In organizations it often becomes embedded not only in documents and repositories but also in organizational routines, processes, practices and norms (Davenport et al., 1997).

Following this definition, it is arguable that knowledge is not static, rather, it is dynamic and changing over time, according to the new discoveries and new ideas as they evolve. In this context, ideas can be seen as the mechanism by which

the existing knowledge pool is updated. The outputs of idea detection could be considered as part of the knowledge in the DIKW pyramid. In comparison, the outputs of ideation (i.e. the creation of new ideas) can be classified into two categories: the ones that fit into existing knowledge and those that are to be discarded. Generated ideas that are discarded might be reconsidered at a future time, when new facts are discovered or new technologies become available, that make these ideas practicable and acceptable. The newly evolved technologies or concepts that make the novel ideas become true are termed as *catalysts*. Examples of the knowledge changes caused by new discoveries are shown in the Table 1.2:

TABLE 1.2: Examples of how new discoveries change the knowledge pool

old knowledge	new knowledge	catalysts
Continental drift is not acceptable	Continental drift is acceptable	New discovery of plate tectonics in the 1950s
The concept of evolution is not acceptable	The concept of evolution is acceptable	New discovery of a mechanism by which it could work in the idea of natural selection, which is inspired by economics - an example of cross-disciplinary ideation

Examples of the knowledge changes caused by novel ideas are shown in the Table 1.3:

TABLE 1.3: Examples of how novel ideas change the knowledge pool

old knowledge	new knowledge	catalysts
cup is not edible	cup is edible	novel idea: make cup using biscuit
human cannot have conversations with machines	human can have conversations with machines[3]	novel idea: using artificial intelligence techniques to train the machine to be able to talk

(4)Wisdom. The variations of the definition of wisdom can be reviewed from the following two aspects: (1) the relation between wisdom and other components in the DIKW pyramid; (2) The usefulness of wisdom. One definition of wisdom is described as integrated knowledgeinformation made super-useful (Cleveland, 1982). Rowley credits the following version of wisdom definition to Ackoff, who defined wisdom as the ability to increase effectiveness. Wisdom adds value, which requires the mental function that we call judgment. The ethical and aesthetic values that this implies are inherent to the actor and are unique and personal (Rowley & Hartley, 2008).

Scientific ideas are expected to solve problems in a creative and more effective manner. Detecting and generating novel ideas would eventually contribute to the wisdom, but this is outside of the scope of computation for the foreseeable future.

1.2 Research problem

Although some researchers have been working on the similar areas, there are research gaps need to be tackled. The following research problems are identified:

- There are lacking of enough investigations on the functional definition of idea, which leads to the obstacles to conduct experiments on extracting ideas using computer algorithms.
- Previous research on idea mining is not focusing on scientific publications.
- According to the literature available, State-of-the-art algorithms were not applied in idea mining, such as neural networks.
- Very less literature on novel idea computation.
- Citations as idea containers were not researched from sequential perspective.

To summarize, there are tree major research subareas that need to be focused: idea mining, novel idea computation and idea flow analysis.

1.3 Research objectives

1.3.1 The research goals.

Although idea detection and creation are challenging, the potential impact of these tasks on information retrieval and computational creativity is ubiquitous. The experimental results of this work will show that the idea definition and creation

schemes proposed in this work are effective but with room for improvement. Table 1.4 shows an example approach, illustrating the expected and the actual outcomes of one of the research goals in the original research statement: *idea exists in text, thus, it should be possible to spot the idea patterns using artificial intelligent techniques to detect idea automatically.* specifically, it demonstrates the goal of this research is to explore the accuracy achievable in implementing idea detection algorithms - in this case, using abstracts where **P:** indicates a problem span starts and **S:** indicates a solution span.

Other goals of this research are: automatically predict links between problem and solution statements that haven't occurred together; analyzing citations to observe the idea flow.

To summarize, there are six goals in this research:

Goal 1: formal idea definition. The ideas could not be computed until they were defined with mathematical terms and formulations. The first goal of this work is to define *idea* formally. The idea definition should satisfy the following criteria: first, the defined ideas are understandable by human; second, they are extractable by computer algorithms; third, they exist in text.

Goal 2: novel idea computation. A framework for automatically computing novel ideas is needed. For each step in the framework, certain functions are required to be defined clearly. Investigate strategies to evaluate of the computed ideas.

Goal 3: human annotated dataset construction. In order to use machine learning techniques to extract ideas automatically, human annotated dataset needs to be collected. The first step is to get together scientific text corpus that are available. The second step is to ask readers to hight ideas in the text corpus.

Goal 4: applying neural network to idea detection. Since neural network has been proved to be an effective machine learning algorithm, it would be interesting to explore specific neural network technique on detecting ideas from scientific text.

TABLE 1.4: The expected and the actual outcomes of the research goal: idea detection from abstracts.

Abstract with expected idea	Automatically detected ideas
Simultaneous Pursuit of Sparseness and Rank Structures for Matrix Decomposition. In multi-response regression, ***P:* pursuit of two different types of structures is essential to battle the curse of dimensionality**. In this paper, we seek a sparsest decomposition representation of a parameter matrix in terms of a sum of sparse and low rank matrices, among many overcomplete decompositions. On this basis, we propose a ***S:* constrained method subject to two non-convex constraints, respectively for sparseness and low-rank properties**. Computationally,obtaining an exact global optimizer is rather challenging. To overcome the difficulty,we use an alternating directions method solving a low-rank subproblem and a sparseness subproblem alternatively, where we derive an exact solution to the low-rank subproblem,as well as an exact solution in a special case and an approximated solution generally through a surrogate of the L_0-constraint and difference convex programming,for the sparse subproblem. Theoretically,we establish convergence rates of a global minimizer in the Hellinger-distance,providing an insight into why pursuit of two different types of decomposed structures is expected to deliver higher estimation accuracy than its counterparts based on either sparseness alone or low-rank approximation alone. Numerical examples are given to illustrate these aspects, in addition to an application to facial imagine recognition and multiple time series analysis.	***S:* Simultaneous Pursuit of Sparseness and Rank Structures for Matrix Decomposition**. In multi-response regression, pursuit of two different types of structures is essential to battle the curse of dimensionality. In this paper, we seek a ***S:* sparsest decomposition representation of a parameter matrix in terms of a sum of sparse and low rank matrices, among many overcomplete decompositions**. On this basis, we propose a ***S:* constrained method subject to two non-convex constraints, respectively for sparseness and low-rank properties**. Computationally,***P:* obtaining an exact global** optimizer is rather challenging. To overcome the difficulty,we use an ***S:* alternating directions method solving a low-rank subproblem and a sparseness subproblem alternatively, where we derive an exact solution to the low-rank subproblem,as well as an exact solution in a special case and an approximated solution generally through a surrogate of the L_0-constraint and difference convex programming,for the sparse subproblem**. Theoretically,we establish convergence rates of a ***S:* global minimizer in the Hellinger-distance,providing an insight into why pursuit of two different types of decomposed structures is expected to deliver higher estimation accuracy than its counterparts based on either sparseness alone or low-rank approximation alone**. ***P:* Numerical examples are given to illustrate these aspects, in addition** to an application to facial imagine recognition and multiple time series analysis.

Goal 5: analyzing the idea containers. Citation sentiment can be used as a signal of the current view of the idea it denotes. In order to analyze citation sentiment automatically, word2vec model based on scientific publications is worth to try.

Goal 6: Analyzing citation sentiment sequences. According to my knowledge, idea flow is a novel research concept. Here, I define idea flow as the citation sentiment sequence. To sense the idea flow, citation sentiment sequences need to be further studied, both from internal and global perspective. Algorithms to analyze the citation sequences need to be invented and tested.

1.3.2 Contributions.

The contributions of this work are as follows:

- The adaptation of a functional formal definition of an idea.
- The development of new rule based methods (by firstly collecting cue words to classify phrases into solution phrases) for detecting idea pairs.
- Establishment of new measurements for the evaluation of ideas, which are compared against each other.
- A novel approach for generating brand new ideas; A technique which is both explained, explored and evaluated.
- Collection of first corpus with human-annotated idea labels; The dataset was collected in order to provide a ground truth for machine learning approaches to idea detection [4].
- Application of a neural network methodology to automatically label ideas, which is based on the dataset detailed above.
- An analysis of the extent to which ideas can be signaled by collocation with internal document constructs such as citations.

[4]The quality of the human annotated idea pairs need to be evaluated in the future.

1.4 Research scope

This research is limited to scientific publication abstracts for the following reasons:

- There are literatures studying ideas extracted from web pages, as well as mining social ideas, however, there are not many publications on mining ideas from scientific text.
- Abstracts in publications are more accessible in comparison with the whole publications.
- Abstracts contain concentrated information, which has more value.

1.5 Structure of the rest of the thesis

The rest of the work are organized as follows:

- Chapter 2: **Literature Review.** The literature review investigates relevant research papers that have been published, covering data mining, machine learning, the basic techniques of natural language processing (NLP), sentiment analysis and computational creativity.
- Chapter 3: **Research Methodology.** Datasets for the experiments are explained in section 3.1. Experiments are conducted under both existing methods and the proposed novel methods, which are detailed in section 3.2.
- Chapter 4: **Results and Discussion.** The experiments results are discussed in this chapter. Both the positive and negative perspectives of each method are discussed. Possible reasons that may lead to the results are discussed as well.
- Chapter 5: **Conclusion and Future work.** This chapter concludes the contributions of this work as well as describes the possible extensions of this work. Potential future work that will emerge is envisaged.

The road map of this work is demonstrated in Figure 1.2. The tasks carried out in this work are demonstrated on the left and the techniques for each task are listed on the right. Among the techniques, Natural language processing (NLP) is the most important part for all the tasks. Several detailed techniques, such as part-of-speech tagging (Schmid, 1994) and dependency tree parsing (Mcdonald et al., 2005), are used through all the tasks. Computational creativity and collaborative filtering are used for idea computation, where computational creativity guides the process to brainstorming and collaborative filtering is the key method for finding links between different concepts. In the task of dataset construction and idea detection using supervised learning fashion, two techniques play an important role, which are deep learning and named entity recognition. Ideas are labeled as special entities, which are to be classified by deep learning. In the training process, Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 2013), conditional random field (CRF) (Zhao et al., 2006) and word2vec (Mikolov, Chen, et al., 2013) worked together for learning the model. Word2vec is also used in argumentative-zoning and idea sentiment experiments. Idea sentiment study in this work is based on the study of citation sentiment detection. In idea flow study, sentiment level co-occurrence matrix is implemented and tested. Eddy dissipation rate is also one of the methods for analyzing idea flow.

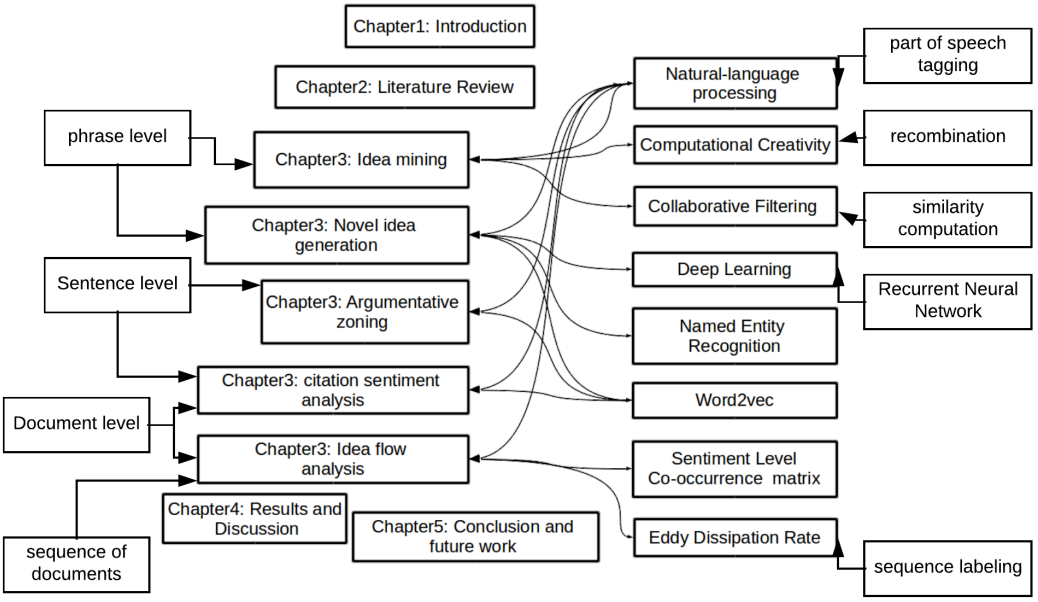


FIGURE 1.2: Road map.

Chapter 2

Literature Review

To summarize, idea mining, novel idea computation and idea flow analysis brings together the following fields:

- Idea mining: *data mining, machine learning, natural language processing*
- Novel idea computation: *computational creativity, information retrieval*
- Idea flow analysis: *Word2vec, argumentative zoning, sentiment analysis, texture analysis in image processing, Eddy Dissipation Rate*

2.1 Idea mining

Unique and high-quality creative ideas are usually generated via brainstorming. The research of brainstorming has been well studied for last several decades. (Gallupe et al., 1992; Rossiter & Lilien, 1994; White et al., 2012). However, due to the information overload, it is difficult to create novel concepts efficiently manually. Therefore, we need computer as an assistant for generating novel ideas. The concept of Computer-Assisted Brainstorming (CAB) was established by Hollander (Hollander, 1984) in the 1980s, and envisioned interactive computer programs designed to enhance creative thinking. It was several decades later, however, before researchers successfully developed software tools to support brainstorming.

Hardenberg *et al.* (von Hardenberg & Bérard, 2001) introduced a *Bare-hand HCI* system, which integrated optical finger tracking into a two-phase brainstorming scenario. Phase 1 involved the collection a large number of ideas from participants and display on a video wall, with phase 2 seeing participants freely and simultaneously rearranging these items via touch manipulation. More recently, Biemann *et al* (Biemann et al., 2004, 2006) developed *SemanticTalk*, software for visualizing brainstorming sessions and thematic concept trails that acted as a visual memory with both spoken dialogs and text documents being captured on a two-dimensional plane.

One of the key features of *SemanticTalk* was its ability to automatically generate associations between terms within the text identified as being important. This process of identifying key concept terms is being extended by the nascent field of *Idea Mining* (Thorleuchter, 2008), which focuses on the task of extracting reified idea structures that are embedded within text - whether that be in websites (C. Wang et al., 2007; Yoon, 2012), patents (Thorleuchter, den Poel, & Prinzie, 2010), databases (Park & Lee, 2011), blogs (Thorleuchter & Van den Poel, 2011a; Thorleuchter, Van den Poel, & Prinzie, 2010a) or scientific literature (Thorleuchter & Van den Poel, 2011b). A general approach for Idea Mining was introduced by Thorleuchter *et al.* in (Thorleuchter, Van den Poel, & Prinzie, 2010b), which defined a technological idea as being represented by a combination of a *purpose* and a corresponding *means*, before going on to semi-automatically discover novel idea patterns in unstructured technological texts. However, most of the publications about idea mining did not touch the area of innovation. They solely focus on discovering existing ideas.

2.1.1 Data mining

Recently, our ability to create and collect data has improved dramatically. How to discover and extract useful data based on the business requirements becomes more and more important. From a database perspective, data mining is defined as a process of nontrivial extraction of implicit, previously unknown and potentially

useful information from data in databases (M. Chen et al., 1996). Data mining plays an important role in many tasks, including the applications in research area and industrial business. People reserve data as digital assets because people believe there are valuable patterns that are encoded in the data. For researchers, raw data is used as observations from nature to carry out scientific experiments. In industry, historical data reflects hidden rules that govern the markets or predict the trends (Fayyad & Uthurusamy, 1996). For example, data mining approaches have been utilized in web data mining research. The work described in (Madria et al., 1999) discussed web data mining from the following aspects: web content mining, web structure mining and web usage mining. There's research focusing on mining ideas from web (Thorleuchter & Van den Poel, 2013). Data mining also proved to be effective in solving biological and environmental problems (YANG & WU, 2006). For example, it has been applied in healthcare (Koh & Tan, 2005), where data mining provides solutions to transform the complex healthcare transactions into useful information for making decisions.

Traditionally, the key activities in data mining revolve around identifying patterns and trends from limited databases. With the advent of big data, data mining principles becomes more prevalent. Big data requires more steps in the data mining process. For example, different data sources with different data format need a flexible processing strategies. Big data process diagram can be demonstrated in Figure 2.1.

In order to mine useful results from data, combining various techniques is important. Different tasks or data mining may require different approaches, and one needs to select an appropriate data mining method according to the type of data in question. It also needs justification on the typical data mining methods according to different types of data. As shown in Figure 2.1, taken from the IBM work (M. Brown, 2004)[1]., big data contains data sources with different formats, need different techniques to process. For example, if a business problem requires data both from transaction activities and news circle, then, techniques dealing with relational database and unstructured data are useful. Integrating multiple

[1] *https* : *//www.ibm.com/developerworks/library/ba* − *data* − *mining* − *techniques/*

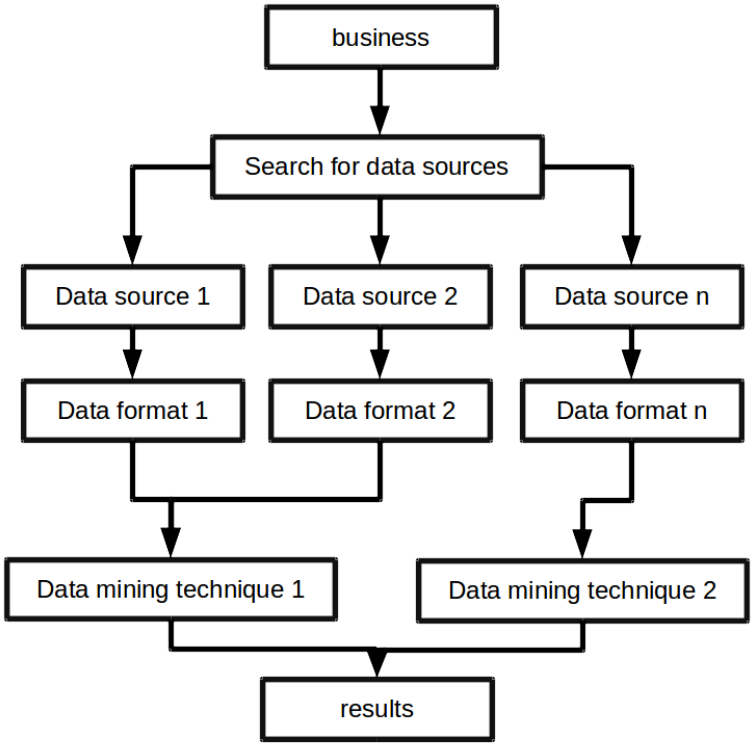


FIGURE 2.1: Business driven data process (M. Brown, 2004). It indicates that different data sources can be structured into different data format. Different data formats may or may not require different data mining techniques

techniques to do research is adopted. For generating novel ideas, bother relational database (storing the problem and solution pairs) and unstructured text data (the raw abstracts from publications) are used. It is challenging to invent novel data mining methods to solve specific problems effectively. The following paragraph reviews some data mining techniques.

Data mining techniques. The survey in the paper (Han, 1996) discussed data mining techniques from the following perspectives: generalization, characterization, classification, clustering, association, evolution, pattern matching, data visualization, and meta-rule guided mining. It also reviews works on data mining techniques from different databases, such as relational, transaction, object-oriented, spatial, and active databases, as well as global information systems. The key techniques are listed bellow[2]:

[2]Some of the reviews are inspired by IBM developerworks *https* : *//www.ibm.com/*

- Association rules

  Association rules are used for discovering interesting relations between variables in large databases. The purpose of learning association rules is to identify relations discovered in databases using some measures of interestingness (Piatetsky-Shapiro, 1991). Association rules have been used in data mining applications to capture relationships present among attributes in large data sets (L. Huang et al., 2014). In association rule mining tasks, people discover correlations between two or more entities. For example, when analyzing people's reading behavior, the hidden reading pattern maybe discovered that a reader who constantly reads machine learning publications probably has interests in big data papers.

  Association rules have been applied in many fields, such as mining rules from transaction records (Agrawal et al., 1993), where all significant association rules between items in the target database are generated using buffer management and pruning techniques. In paper (Fukuda, 1996), an example of bank customers was studied: Age and Balance are two numeric attributes, and CardLoan is a Boolean attribute. Their conclusion is, bank customers whose ages and balances fall in a planar region P tend to use card loan with a high probability.

  Association rules are used in this work in such a way that human collected cue words, such as *problem, solution*, are considered to be the indicators that the real problem or solution phrases are nearby.

- Classification

  Classification (Macqueen, 1967) can be used as a filter engine to keep items that satisfy the predefined criteria. For example, the bank customers can be classified into different groups based on their age, income and other attributes. According to the business problem, only items with specific class are extracted.

  In the tasks of mining text data, classification plays an important role. For example, if the task requires techniques to extract positive reviews from

movie review dataset, it is necessary to use automatic classification skills rather than manually analyze the data. Classification can be applied to segments of text as well. In this study, phrases are required to be classified into problem and solution categories, thus, classification is one of the key techniques to conduct the data processing experiments.

- Sequential patterns

  Sequential patterns are studied in the paper (Agrawal & Srikant, 1995). Each sequence is composed of transactions ordered by transaction time and each transaction is a set of items. In their study, the sequential data mining task is to discover sequences that have the same pattern. For example, one sequential data mining task could be to mine bookshop customers (with a certain percentage) who bought Foundation and Ringworld in one transaction, followed by Second Foundation in a later transaction. Approaches which identify such patterns may well be useful to idea mining. For example, if we see a sequence of words as the representation of problem or solution, then, the task of problem and solution detection could use sequence mining techniques. Another example is, the citation sentiment degrees form a sequence, where there are patterns. The citation sentiment degree changes may have patterns, which can be used for analyzing the attitudes towards a citation.

  In fact, in this study, sequential pattern discovery techniques will prove useful to identify idea trends by analyzing a sequence of citations. For example, if it is useful to mine papers that got cited positively [3] last year and negatively this year, which may indicate that those papers have the ideas that have been improved.

Text mining. Data can be reserved in many devices in many forms. One important form of data is reserved in documents. As the amount of documents have been increased dramatically, mining useful data from them becomes more

[3] Regarding citation sentiment, it's well studies in Athar's thesis: *https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-856.pdf*

and more important and challenging. In general, text mining refers to the process of extracting interesting information and knowledge from unstructured text (Hotho et al., 2005; Franke et al., 2003). Text mining uses some of the techniques from information retrieval, natural language processing (NLP) and connects them with the algorithms and methods of data mining, machine learning and statistics (Hotho et al., 2005).

The related topics such as machine learning and NLP will be reviewed in the following sections.

2.1.2 Machine Learning

While several of the above approaches might be considered as machine learning, the field has a greater focus on prediction than solely mining for patterns. One of the most active research areas in the domain of Artificial Intelligence, the term machine learning, was coined by Arthur Samuel in 1959 (Samuel, 1988). It is a field of computer science that uses statistical techniques to give computer systems the ability to learn with data, without being explicitly programmed. Machine learning is closely related to pattern recognition (Burges, 2008) and computational learning theory (Kearns & Vazirani, 1994) in artificial intelligence (Russell & Norvig, 2010). One of the tasks in machine learning is to make predictions based on observations ("Glossary of Terms", 1998), which is beyond static rule based programming. It is usually built on an amount of labeled data.

The applications of machine learning include the tasks in computer vision and natural language processing. For example, image classification (Foody & Mathur, 2004) and language translation (Tenni et al., 2002) involve data collection, model training and evaluation, which are necessary steps in machine learning.

In general, machine learning can be categorized into supervised learning, semi-supervised learning and Unsupervised learning. The following contents explain the details of these methods.

Supervised learning. One of the most important machine learning tasks is supervised learning. The goal of supervised machine learning task is to learn a function that maps an input to an output based on input-output pairs (Pollack, 1995). Usually, the input-output examples are labeled datasets. Each example is composed of an input object and a desired output value, where the input object could be represented by vectors. Supervised learning algorithms examine the training datasets (Mohri et al., 2012) and generate an inferred function, which performance is controlled by parameters. The learned function is expected to predict the outcome based on new inputs that have not been seen in the training procedure.

Caruana and Niculescu-Mizil (Caruana & Niculescu-Mizil, 2006) analyzed ten machine learning methods including: support vector machines (SVMs), neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees, and boosted stumps. They used of a variety of performance criteria to evaluate the learning methods. Caruana et al. (Caruana et al., 2008) compared these methods when the number of features increases. They also investigated the performance of these algorithms on solving big data problems. For example, there are tens of thousands of attributes in genome problems, and there could be more attributes in text mining problems. According to their study, linear regression and ensemble algorithms have similar performance as the number of attributes grow. Linear regression is on the top three for 5 of the 11 problems in the experiments. Penalized linear regression is a derivative of ordinary least squares (OLS) regression (Bowles, 2015), which is a method developed by Gauss and Legendre almost 200 years ago. There were designed to overcome some limitations of OLS regression. One of those issues is that sometimes it has over-fitting problem. Penalized linear regression can reduce the freedom to some extent. This method is effective to solve problems with many degrees of freedom, such as genetic issues, which have tens of thousands of dimensions. Ensemble methods are to construct different predictive models and combine the outputs by averaging the outputs or chose the majority answer.The individual model is called

base learner. Ensemble methods are based on the assumption that if the individual learners perform slightly better than randomly guess, the integrated ensemble performance from all the individual learners could be a very good classifier.

Although penalized linear regression models and ensemble methods have their unique advantages, the mostly adopted classifier is SVM (Ukil, 2002). The support-vector network is a learning machine for two-group classification problems. It is implemented based on the idea that the input vectors are non-linearly mapped to a very high dimension feature space (Cortes & Vapnik, 1995), where a linear decision surface is built. The decision surface has high generalization learning machine ability, which was achieved by utilizing polynomial input transformations.

SVM is useful for many tasks, such as classifying tissue samples(Furey et al., 2000), image segmentation (Barghout, 2015) and text classification (Tong & Koller, 2000). SVM is an effective machine learning technique for labeled datasets. However, when data are not labeled, SVM will fail to do classification. In this case, unsupervised learning is needed.

In this work, in order to identify ideas, supervised learning is a well suited method in such a way that as long as there are enough human annotated data, the parameters in the objective function can be learned automatically.

Unsupervised learning. In supervised learning, the parameters in the inferring function can be learned by the input-output pairs. In unsupervised learning, the goal is to learn a function that is able to capture the hidden structure from unlabeled data(Barlow, 2002).

Methods to unsupervised learning include clustering (Jain, 1999), anomaly detections (Chandola et al., 2009), neural networks (Haykin, 1994) and learning latent variable models (C. N. J. Yu & Joachims, 2009).

Clustering is the unsupervised classification of patterns into clusters. Clustering is to organize unlabeled data into similarity groups, which are called clusters [4]. A cluster is a collection of data items which are similar between them, and dissimilar

[4] $http://www.mit.edu/\ 9.54/fall14/slides/Class13.pdf$

to data items in other clusters (Hastie et al., 2009). clustering procedure can be described in Figure 2.2.

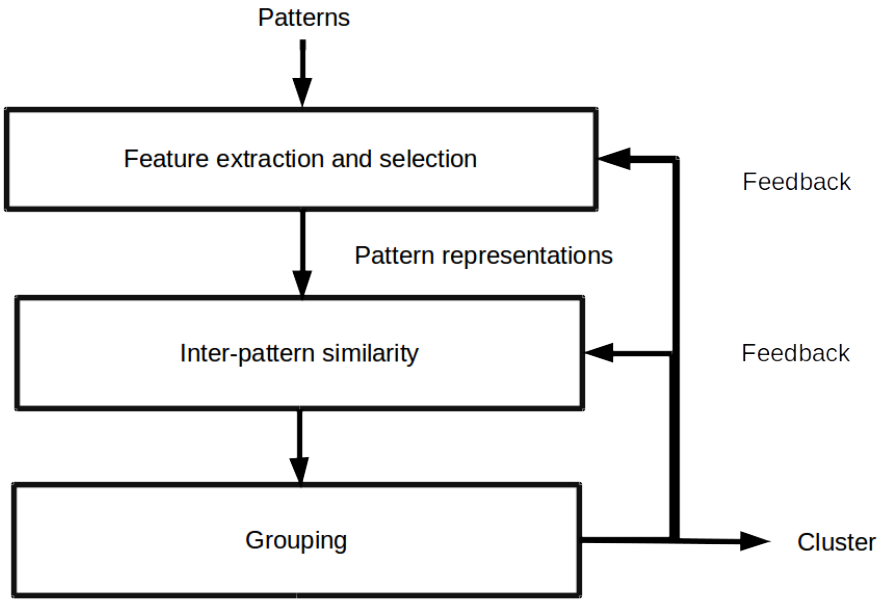


FIGURE 2.2: Clustering stages (Rosenberg et al., 2005).

The outputs of the clustering can be applied in many tasks, such as analyzing global scale patterns in biosphere processes (Elliott et al., 2013) and text mining (Berry, 2004; Larsen & Aone, 1999). One of the most popular method in clustering is k-means (Mohanavalli et al., 2011), which is based on vector quantization.

Artificial neural networks (ANNs) or connectionist systems are computing systems inspired by the biological neural networks that constitute animal brains (Van & Bohte, 2017). ANN learn how to solve problems by examining examples. ANN is different from task-specific programming. ANN can be used to extract features to represent instances. These feature are further used to cluster the data. ANN can also be employed to determine the number of clusters (Erilli et al., 2011).

Latent variable models include Gaussian mixture models, hidden Markov models, and latent Dirichlet allocation, which can be implemented by tensor structure in their low-order observable moments. In detail, parameter estimation is reduced

to the problem of extracting a certain (orthogonal) decomposition of a symmetric tensor derived from the moments. This decomposition can be viewed as a natural generalization of the singular value decomposition for matrices (Anandkumar et al., 2012). Latent Semantic Analysis has many applications, such as information retrieval, natural language processing and text mining (Larsen & Aone, 1999).

Semi-supervised learning. In supervised learning, only data with ground truth can be used for training. It is often difficult to collect enough labeled instances to train the classifiers. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers (Xiaojin, 2007). Even though semi-supervised learning does not require as much time in labeling the training data in supervised learning, a lot of efforts are needed to design good models for semi-supervised learning.

There are several well-known methods to conduct semi-supervised learning including Expectation-Maximization (EM) with generative mixture models (DebashisKushary, 1997; Diplaros et al., 2007), self-training (Rosenberg et al., 2005), co-training (Blum & Mitchell, 1998), transductive support vector machines (Chapelle et al., 2006), and graph-based methods (Bryant, 1986). Which method to choose depends on the business problem and the properties of the available data. It is difficult to judge which methods suits the situation. If the classes give a well clustered data then, EM with generative mixture models maybe a good solution. If the features extracted from the data easily split into sets, then, co-training could be a better option. If the two points with similar features tend to be in the same class, then use graph-based approach. Self-training is also called incremental semi-supervised training. The procedure in self-training is using a combination of weakly and fully labeled data. Weakly labeled data is obtained in such a way that a probability distribution over labels is taken into account when trying to assign the label to unlabeled data. The demonstrations of the batch training approach with EM and incremental self-training are shown in Figure 2.3 and 2.4.

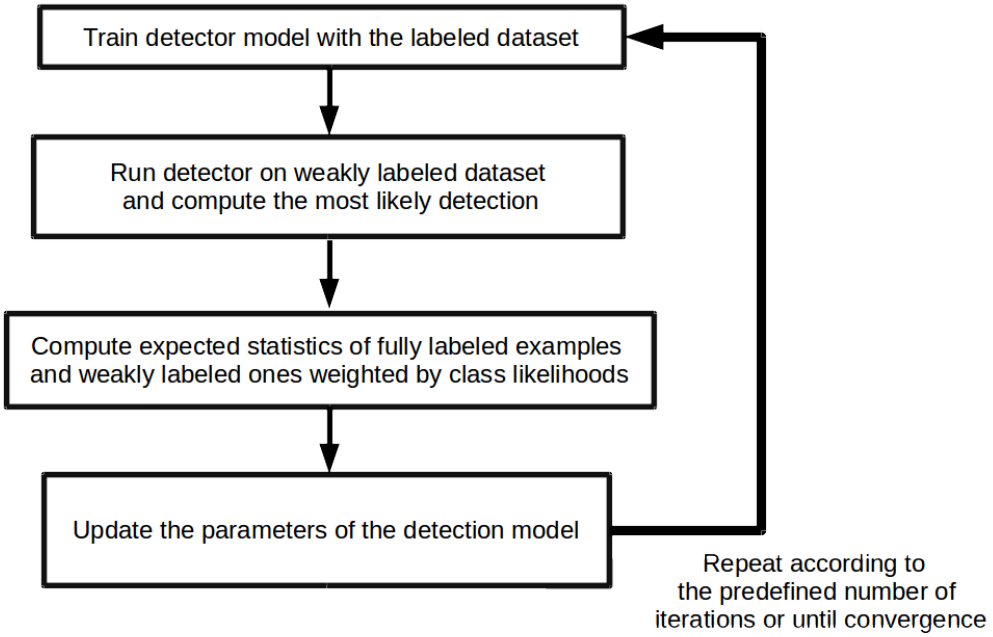


FIGURE 2.3: Schematic representation of the batch training approach with Expectation-Maximization (EM) (Rosenberg et al., 2005).

Deep learning. Deep learning model design is inspired by biological nervous system, which intended to discover the relationship between various stimuli and associated neuronal responses in the brain (Olshausen & Field, 1996). The elements of an artificial neural network include neurons, connections, weights, propagation function and learning rule.

There are several ways to update the weights in backpropagation process. One of the popular method is via stochastic gradient descent (Ketkar, 2014; Bottou, 2010).

There are several famous deep learning architectures, such as recurrent neural networks (RNN) (R. J. Williams & Zipser, 1990) and convolutional neural network (CNN) (Lecun, n.d.). A recurrent neural network has the following features: the connections between units are formed into a directed graph; It has temporal behavior for a time sequence; The internal state in RNN can be used to analyze

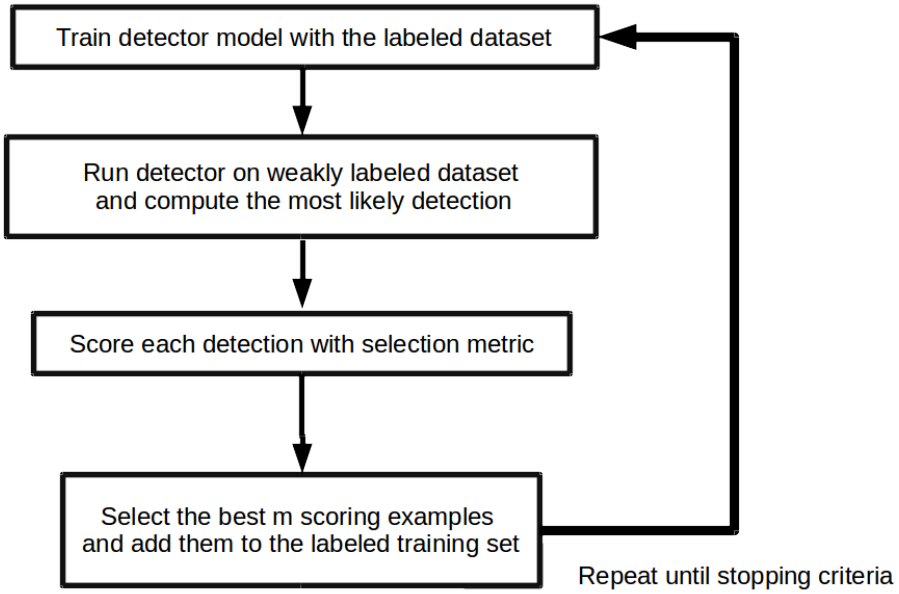


FIGURE 2.4: Schematic representation of the batch training approach with incremental self-training (Rosenberg et al., 2005).

the input sequences. CNNs use a variation of Multi-Layer Perception, which require minimal preprocessing (Lecun, n.d.). CNNs are also called shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics (Casasent & Neiberg, 1995; Ueda & Togawa, 1993). CNNs took the idea in biological processes that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field,, which is called the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use less pre-processing compared to other image classification algorithms, which indicates that the network learns the filters itself. Those learned filters are usually hand-engineered in traditional computer vision algorithms. This does not require prior knowledge and human effort in feature extraction is a big advantage.

The applications of RNN include time-series processing (Inakazu et al., 2016) and emergency event detection (Cortez et al., 2018). In time-series processing, the

Pulsed Neural Networks (PNNs) and RNNs are combined in order to enhance time-series processing ability. In emergency event detection, they applied long short-term memory (LSTM) architecture to their event prediction system. The applications of CNN range from image classification (J. Wang et al., 2016) to text classification (Lai et al., 2015). CNN and RNN can work together in such a way that a joint image-label embedding are learned to characterize the semantic label dependency as well as the image-label relevance, and it can be trained end-to-end from scratch to integrate both information in a unified framework. CNN was effective in natural language processing (W. Yin et al., 2017) as well. In that paper, CNN is used to extract position-invariant features and RNN is used for modeling units in sequence.

Transfer learning. One of the machine learning techniques is transfer learning, which utilizes the knowledge gained through solving one issue and applies it to a different but related problem (Pan & Yang, 2010). For example, the image recognition model learned from daily life image dataset can be reused in medical image classification. In paper (Y. Yu et al., 2017), they took use of VGGNet (Ke et al., 2018) as pre-trained model for their experiments, which enabled them to achieve better results.

Transfer learning is also used in text processing tasks. For example, in the task of detecting stance from Tweets (Zarrella & Marsh, 2016), Zarrella et al. trained embeddings of words and phrases with the word2vec method, then used those features to learn sentence representations and these sentence vectors were then fine-tuned for stance detection on several hundred labeled examples. They concluded that the result was a high performing system that used transfer learning to maximize the value of the available training data.

This study also used transfer learning to learn sentence representations, which were used to detect phrases as well as predict idea sentiment. Specifically, the representations were learned from different data sources, one is from scientific papers and the other is from google news.

2.1.3 The fundamental techniques of NLP

NLP plays an important part in this research. The data to be analyzed is unstructured, thus, it is necessary to use NLP techniques to process, clean and analyze the data according to research goals.

There are basic techniques in NLP, including tokenizer, part-of-speech-tagging, representation of meaning, semantic similarity and language modeling.

Tokenizer. In the early stage of text processing, the raw text is input into the tokenizer (He & Kayaalp, 2006; Pentheroudakis et al., 2006), which breaks up the character stream into individual words (tokens). For example, the piece of text *in this paper* would be broken into the terms [in, this, paper] using whitespace tokenizer.

In this work, Python based Natural Language Toolkit (NLTK) provides the tokenizer[5] for the experiments.

Part-of-speech-tagging. One of the basic techniques in NLP is part-of-speech tagging, also called POS tagging. It is the process of labeling a word in a piece of text as corresponding to a particular part of speech, based on both its definition and its context. An example of the output of POS tagging is demonstrated bellow:

The [determiner] smooth [adjective] endoplasmic [adjective] reticulum [noun] also [preposition] carries [verb] out the [determiner] metabolism [noun] of [preposition] carbohydrates [noun] and [conjunction] steroids [noun].

This results are generated using A Part-Of-Speech Tagger (POS Tagger)[6], which is a piece of software that automatically assigns parts of speech to the input text. This tool is based on the log-linear part-of-speech tagger algorithm described in the paper (Manning & Manning, 2000). In the process of POS tagging, every word in the input text is expected to be marked with one of the categories, such as noun, verb, adjective, adverb, etc. Take the above sentence for example, the words

[5] *http : //www.nltk.org/api/nltk.tokenize.html*
[6] *https : //nlp.stanford.edu/software/tagger.shtml*

The, smooth, endoplasmic and reticulum were marked as *determiner, adjective, adjective and noun* respectively.

POS tagging technique has been investigated both from implementation and application perspective. Paper (Brill, 1992) introduced a simple rule-based part of speech tagger, which automatically acquires its rules and tags with accuracy coinparable to stochastic taggers. It was also stated in that paper that the rule-based tagger has many advantages over stochastic taggers, such as, a reduction on information that is needed, ease of improving the tagger being utilized, and better portability from one dataset to another. Following this idea, some extensions to this rule-based tagger were explored. For example, in the paper (Manning & Manning, 2000), a new method was described for expressing lexical relations in tagging that are not captured by stochastic taggers. Manning et al. also showed that a rule-based approach is capable of tagging unknown words and proved that multiple tags can be assigned to words in some cases of uncertainty.

POS tagging has been adopted in several NLP tasks. For example, automatically tagging manuscript (Bollmann et al., 2013), improving summarization performance (Manne & Sameen Fatima, 2013). POS tagging has proved to be an effective technique in the above applications. However, some findings showed that POS tagging is not appropriate for some tasks. In paper (Foster et al., 2011), POS tagger was used to supply the input to Malt (Nivre et al., 2006), which is a statistical dependency parser. The dependency tree was misparsed due to the errors in POS tagging.

In this work, in order to generate problem and solution concepts, noun words need to be detected. The first method being explored is POS tagging, which is used for labeling noun phrases.

Representation of meaning. The representation of meaning of expressions in a natural language has been studied for decades. Some researchers used Possibilistic Relational Universal Fuzzy (PRUF) (Zadeh, 1977) to design a novel ftype of synthetic language, for the purpose of setting up a target language for representing meaning. In paper (Rumfitt, 1994), system of graphs was discussed. The author

came to a conclusion that anaphoric and relative pronouns are structural signs not linguistic constituents. Certain count nouns are made the basis of this system. Verbs and Nouns are important elements in a sentence, thus, verb and-noun pairs have widely been used to represent the meaning of an argument as the set of its governing predicates, or, conversely, to represent the meaning of a predicate (Lenci et al., 2002; Hui, 2001). In the section 3.2.1 Chapter 3, nouns are extracted to represent the problem and solution statements in an abstract.

N-grams (Banerjee & Pedersen, 2003) has been widely used as features for text analysis. But some n-grams is not easy to implement (Sidorov et al., 2014). With the advent of neural networks (G. E. Hinton et al., 2006), it is possible for computers to learn feature representations automatically. Recently, word embedding technique (Mikolov, Chen, et al., 2013) has been widely used in NLP community. There are plenty of cases where word embedding and sentence representations have been applied to short text classification (P. Wang et al., 2015) and paraphrase detection (Socher, Huang, et al., 2011).

The tool of word2vec proposed by Mikolov et al. (Mikolov, Chen, et al., 2013) has gained a lot attention recently. With word2vec tool, word embeddings can be derived from big amount of text corpus and the semantic relationships between words can be measured by the cosine distances between the vectors. The idea behind word embeddings is to use distributed representation (M. J. Hinton Geoffrey & Rumelhart, 1986) to map each word into k-dimension vector. How these vectors were computed? The common method to derive the vectors is using neural probabilistic language model (Bengio et al., 2003). The underlying word representations for each word are obtained while training the language model. Similar to the mechanism in language model, Mikolov et al. (Mikolov, Chen, et al., 2013) introduced two architectures: Skip-gram model and continuous bag of words (CBOW) model. Each of the model has two different training strategies, such as hierarchical softmax and negative sampling. Both these two models have three layers: input, projection and output layer. The word vectors are obtained once the models are optimized. Usually, this optimizing process is done using stochastic gradient descent method. Labels are not required when training the models, which makes

word2vec algorithm more valuable compared with traditional supervised machine learning methods that require a big amount of annotated data. Given enough text corpus, the word2vec can generate meaningful representations.

The generated embeddings can be used in many tasks, such as machine translation (Garcia et al., 2017), parsing (Hsieh & Ma, 2016) and semantic search (Bollegala et al., 2015). Word embeddings have two important qualities: they can be trained from unlabeled data and they can be used for a variety of tasks. Word embeddings (word2vec) have been applied to meaning representation learning (Mikolov, Sutskever, et al., 2013; E. H. Huang et al., 2012; Turney & Pantel, 2010; Turian et al., 2010), multi-task learning (Collobert & Weston, 2008), machine translation (Zou et al., 2013; Garcia et al., 2017), parsing (Hsieh & Ma, 2016; Socher, Lin, et al., 2011), semantics (Grefenstette et al., 2013; Zanzotto et al., 2010; Mitchell & Lapata, 2010; Bollegala et al., 2015), sentiment analysis (Maas et al., 2011; Yessenalina & Cardie, 2011; Tang et al., 2014; Xue, Fu, & Shaobin, 2014; Zhang et al., 2015), topic modeling (Larochelle & Lauly, 2012), paragraph and document modeling (Socher, Huang, et al., 2011; Srivastava et al., 2013) and text classification (Lilleberg et al., 2015). Despite the variety of word embedding usage, there is no concrete rule that guides us how to combine word embeddings for a sentence of a paragraph. Sadeghian and Sharafat (Sadeghian & Sharafat, n.d.) explored averaging of the word vectors in a sentiment review statement. Their results indicated that word2vec models significantly outperform the vanilla bag-of-words model. Amongst the word2vec based models, softmax provides the best form of classification. Tang et al. (Tang et al., 2014) used the concatenation of vectors derived from different convolutional layers to analyze the sentiment statements. They also trained sentiment-specific word embeddings to improve the twitter sentiment classification results. In (Kenter et al., 2016), word embeddings were optimized for sentence representations. They computed sentence embeddings by averaging word embeddings with an optimization scheme that is specifically for the purpose of averaging. In this chapter, a method to learn sentence embeddings for the task of AZ was explored. The results were analyzed from three aspects: the impact of the

training corpus, the effectiveness of specific word embeddings and different ways of constructing sentence representations based on the learned word vectors.

Semantic similarity. Machine can help people to estimate the similarity between things, which maybe represented by words, sentences, or concepts (they maybe further converted to vectors) (Sunkar, 2015). Semantic similarity is a metric for estimating the distance between the two representations. It only includes *is a* relations (Ballatore et al., 2014). Some measures combine several approaches, for example, in paper (Resnik, 2012), edge-based and node-based approach were integrated for information content calculation.

In text mining, Term frequency-inverse document frequency (TF-IDF) is often used as a weighting factor. TF-IDF weights have been used in classifying documents by measuring appearance frequency of each keyword in a document. Most of the text-based recommender systems in the domain of digital libraries use TF-IDF (Beel et al., 2015).

Probabilistic language modeling. The goal of probabilistic language modeling is to compute the probability of a sentence or sequence of words. An example of probability computation for the sentence $S = The\ man\ is\ feeding\ the\ dog.$ is demonstrated bellow:

$$P(S) = P(The, man, is, feeding, the, dog)$$

where P(The, man, is, feeding, the, dog) can be computed using chain rule (Ambrosio & Dal Maso, 1990):

$$P(The, man, is, feeding, the, dog) =$$
$$P(The)P(man|The)P(is|The, man)P(feeding|The, man, is)$$
$$P(the|The, man, is, feeding)P(dog|The, man, is, feeding, the)$$

This techniques is used in computing the word vectors, where word2vec is used in this work to evaluate citation sentiment.

2.1.4 Some NLP applications

Named Entity Recognition (NER). Named entities are phrases that contain the names of certain cagegories, such as person, organization and location (T. K. Sang & Erik, 2002). Named entity recognition (NER) falls into the task of information extraction. One example is shown bellow:

[ORG U.N.] official [PER Ekeus] heads for [LOC Baghdad].

This sentence has three named entities: Ekeus is a person, U.N. is an organization and Baghdad is a location.

The methods for the task of NER can be grouped into supervised (e.g.: Hidden Markov Models (HMM) (Bikel et al., 1998),Decision Trees (Sekine, 1998) and Maximum Entropy Models (Borthwick et al., 1998)), semi-supervised (e.g.: using regular expressions (Brin, 1998)) and unsupervised (e.g.: based on WordNet (Alfonseca & Man, 2002)). Conditional Random Fields (CRFs) is one of the widely used supervised learning method. In the paper (Lafferty et al., 2001), CRFs was presented as a framework for building probabilistic models to segment and label sequence data. As one of the undirected graphical models, CRFs have shown empirical successes in Part-of-speech tagging (Lafferty et al., 2001) and noun phrase extraction (Fei & Pereira, 2003). There are several ways to use CRF for NER. In the paper (Collobert et al., 2011), CNN and CRF were used together. In the paper (Z. Huang et al., 2015), a model based on hand engineered features using CNN was presented. The paper (D. Lin & Wu, 2009) and (Passos et al., 2014) used linear chain CRF together with hand-crafted features for the task of NER. The paper (Lample et al., 2016) used letter-based representations and bi-LSTM for NER.

The features that have been used for NER include word-level features, document and corpus features and List lookup features. The mechanism in NER can be applied in idea detection. Since idea was defined as <problem, solution> pair, where *problem* and *solution* are represented by a list of words, it is reasonable to associate *problem* and *solution* with two different types of entity.

In this work, NER concept is used for detecting ideas using neural network techniques in such a way that problem and solution are seen as special entities.

2.2 Novel idea computation

While the systems described in 2.1 offer valuable digital support for the iteration component in real-world brainstorming, they are all limited in one important respect: they still rely heavily on human input to generate the novel ideas themselves. A possible way to attack this issue is to generate new links between problems and solutions - and a plausible approach to doing this is to harness the success of collaborative filtering (CF) techniques.

For a computer, it is easy to generate tens of thousands of idea candidates. But it is not easy to judge which ideas are valuable or have potential to impact the world in the near future. Evaluating generated ideas is not the same as evaluating classification results in traditional machine learning tasks. Usually, the objects to be classified have static and fixed labels. For example, in image recognition (Tambm publicado como CN1685371A & Bell, n.d.), the label of each image will not change over time. In comparison, the label of idea is changing due to the change of the context. For example, the idea of applying self-driving devices in the activity of farming was questioned several decades ago (Grosse, 1992), but self-driving farming may become very popular once people make significant breakthrough on self-driving technology (Bojarski et al., 2016). According to my own idea, one possible way to evaluate the novel idea computation system is by splitting the entire dataset into two parts according to date of release, with the dataset that was published later being used for testing.

The related fields are computational creativity and recommender system, which are reviewed in the following subsections.

2.2.1 Computational Creativity

A challenging stage in this work is to generate novel ideas automatically. The techniques for idea detection are not enough for generating new ideas. Idea generation requires knowledge from computational creativity.

The main tasks in Artificial Intelligence (AI) research are reasoning, planning, learning, natural language processing, perception and the ability to move and manipulate objects[7]. Computational creativity, as one of the aims of AI, is a growing area of research (Besold, Schorlemmer, & Smaill, 2015; Besold, Khnberger, & Veale, 2015). Creativity can be defined as the ability to generate novel and valuable ideas (Boden, 2009). As stated in Boden's paper (Boden, 2004), novel ideas maybe produced by combination, exploration or transformation. Ideas cover many aspects, such as arts, music, culinary and linguistic, etc. In general, ideas can be represented by the integration of concepts. Novelty refers to being new, original, or unusual. Valuable means useful and interesting. Modern research in computational creativity has resulted in fruitful theories and systems. Colton et al. (Colton et al., 2011) proposed the first draft of computational creativity theory (CCT), introducing two models: FACE model for describing creative acts of software and IDEA model for measuring the impacts of the creative acts. This theory provided guidelines for computational creativity system design.

FACE is the abbreviations of the four phrases: Framing information, Aesthetic measures, Concepts and Expression of a concept. The creative act should be able to generate these four elements. By the word concept, they referred ton an executable program, which is capable of taking input and producing output. By the phrase expression of a concept, they meant an instance of an (input, output) pair produced when the concept's program is executed. By the term aesthetic measure, they meant a function which takes as input a (concept, expression) pair one of which can possibly be null and outputs a real value between 0 and infinity. By an item of framing information, they meant a piece of natural language text that is comprehensible by people, which refers to a non-empty subset of generative

[7] $https://en.wikipedia.org/wiki/Artificial_intelligence$

acts in the tuple. Such framing information may be domain-specific, and adds value to the generative acts, possibly by (1) putting them into some cultural or historical context, or (2) describing the processes underlying the generative acts, or (3) providing calculations about the concepts/expressions with respect to the aesthetic measures, or (4) obfuscating the creative process and/or the output produced, in order to increase the amount of interpretation required by audience members. By allowing exactly one or zero of each type of generative act, the above definition of a creative act includes singular generative acts. Such acts include those performed by machine learning systems which generate a single classifying concept, and those performed by constraint solvers, where a single expression (the solution) of a concept (the constraint satisfaction problem) is produced. This work follows this creative act model in such a way that the concept *idea* is concretely defined, a system is able to produce novel ideas based on the input old-ideas, framing information is making sure that the outputs are understandable by human, and the aesthetic measure is created based on the measurements in information theory.

IDEA is the abbreviations for the following four words: Iterative, Development, Execution and Appreciation. This cycle provides the guide for making softwares. This IDEA model is expected to capture the notion that the fine-tuning of software has an effect on the impact of its creative acts. The authors of the IDEA model wish to capture the notion that the novelty of the results of a creative act can affect people's appreciation of those results. In this work, the computed novel ideas can be input into the novel idea computation system and iteratively generated more ideas. However, there are lacking of human evaluation on the outputs, thus, if the outputs affect people's appreciation or not needs to be further investigated, which is out of this study scope. To emphasize, this study adopts precision and recall measurements to evaluate the outputs.

Cook et al. (Cook et al., 2012) designed a system, A Novel Game-Evolving Labrat (ANGELINA), that can automatically develop games under a defined theme, by selecting appropriate multimedia content from a variety of sources and incorporating it into a games design. Their game design system was evaluated under the

FACE model and proved to have generative capacity under the four aspects. The system ANGELINA is an application of computational creativity in the area of game design, which provided a new perspective on the creative processes. There are also computational creativity systems in other fields, such as music, art and culinary. Pinel et al. (Pinel et al., 2015) built a computational creativity system for culinary recipes. Before generating ingredient combinations, recipes and ingredients knowledge were constructed based on the existing data. Once a combination was selected under the guide of novelty and quality evaluators, the system further recommended ingredient proportions using distributional conformance method and recipe steps were created by sub-graph composition algorithm. Usually, human evaluation is necessary to determine the novelty and usefulness.

Generating creative scientific ideas, as one of the main intelligent human activities, plays a critical role in improving the quality of life. Usually, scientific ideas are documented in publications, from which existing ideas can be extracted and novel ideas are created by integrating or modifying old ideas. From the human behavior point of view, the idea innovation process starts with digesting old ideas, namely, literature review. The amount of accumulated information is too large to process manually. In the era of AI, researchers put a lot of efforts into using computer algorithm to assist the idea innovation process (Altshuller, 1999). Similar to the creativity in culinary, creative scientific ideas can be generated by recombining concepts. To conduct computational creativity using computer algorithms, designers not only have to consider the four aspects in the FACE model (Colton et al., 2011), but also have to satisfy the goals of computational learning theory (Angluin, 1992):

Give a rigorous, computationally detailed and plausible account of how learning can be done.

The works in this study satisfy the above theory in such a way that the scientific idea can be defined as a <problem, solution> pairs and the innovation process can be described as recommending new links between problems and solutions. To make this process computationally plausible, the concepts were expressed by noun

phrases. An evaluation system was introduced using the measurements that are commonly adopted in information retrieval activities.

2.2.2 Information retrieval

Information retrieval (IR) is a task focusing on the following aspects: the structure, analysis, organization, storage, searching, and dissemination of information (Salton & Harman, 2003). The results of IR are usually directed to end users who cares about certain types of information. In practice, IR is expected to retrieve relevant documents based on user query and the available information. The performance of IR should be evaluated by the following factors: consistency, accuracy and speed[8].

IR techniques. Reviewing IR techniques is a big topic, because it covers a variety of areas, such as searching techniques and IR modeling. It is also related to data types and data structures (Rasmussen, 1999), for example, for searching web information, it requires syntax analysis, Knowledge annotation and Metadata based search using RDF (Resource Description Framework) (Lasic-Lazic et al., 2000). There are also several types of IR models, which specify the unique document representation, the query representation and the retrieval functionality (Anwar A, 2010).

- Boolean model

  Boolean model is one of the earliest IR models. It uses the operators from George Boole mathematical logic (Bleich, 1992), query terms and the corresponding sets of documents are formed into new sets of documents. In a Boolean model, a document is represented by a list of keywords and queries are also expressed by a set of keywords, which are connected with each other by AND, OR, or NOT operations. The retrieval scheme in Boolean

[8] $http://www.doc.ic.ac.uk/\ nd/surprise_97/journal/vol2/hks/infor_ret.html$

model labels a document as either relevant or irrelevant. The disadvantage of Boolean model is that it fails to rank the returned list of documents (Hiemstra & Vries, 2017).

- Vector Space Model

  The vector space model has been discussed for decades (Wong & Raghavan, 1984; Soboroff & Nicholas, 2000). Although there are different definitions and expressions of vectors, it is well accepted that vectors and their similarity measurements can be used to decide the relevance between the document and the query. The commonly used vectors are expressed by the vectors of term weights (the terms appear in the document). Usually, the term weights are calculated by term frequency inverse document frequency (tfidf).

  Similarity between documents stored in this form is often measured by cosine similarity between the vectors.

- Probabilistic model

  The concept of using probabilistic approach in information retrieval was discussed by Maron and Kuhns (Maron & Kuhns, 1997; Jones et al., 2000). It is aiming at answering the question: what is the probability that the examined document is relevant to the target query? Both document and query are represented by the same data format, for example, using binary vectors (Hiemstra & Vries, 2017; Salton & Mcgill, 1983). Stephen Robertson (Robertson & Jones, 1976) formulated the probability ranking principle as follows: If retrieved documents are ordered by decreasing probability of relevance on the data available, then the system effectiveness is the best to be gotten for the data.

  This principle indicates that the probability of relevance is not necessarily the end, but it gives a reference to the user.

- Inference network model

  In inference network model, IR is treated as an inference process in an inference network (Blair, 1979). The implementation of this model is described

as follows: a document is initialized by a term with a certain weight. Given a representation of a query, multiple terms are highlighted, therefore, the credit from those terms is increasing. Ranking document is similar to ranking the vectors in the probabilistic models.

Evaluation. To evaluate the performance of IR system, measurements are needed. Precision and recall are commonly used indicators (Raghavan et al., 1989; Davis & Goadrich, 2006). Precision is the fraction of relevant items among the retrieved items, while recall is the fraction of relevant items that are successfully retrieved over the total relevant items.

Collaborative Filtering. Overwhelmed by the available choices, users would like to identify their interested items effectively and efficiently. One solution to this problem is using recommender system as an assistant (Herlocker, 2015). One of the technologies used in recommender systems is collaborative filtering (CF). CF is described as follows (Oldale et al., 2004):

A method of filtering data to predict an observation about an item for a particular case is provided in which: a set of data representing actual observations about a plurality of items for a plurality of different cases is models as a function of a plurality of case and item profiles, each profile being a set of parameters comprising at least one hidden metrical variable, the parameters defining characteristics of the respective case or item; a best fit of the function to the data is found in order to find the values of the item profiles; and the profiles found are used together with the function to predict an observation for a particular case about one or more items for which data is not available for that case.

CF algorithms (Breese et al., 1998; Sarwar et al., 2001) have proven to be extremely effective in scientific research and real-world applications. CF uses the known preferences of a group of users to make recommendations (i.e. predictions) of the unknown preferences for other users (Su & Khoshgoftaar, 2009) and CF techniques generally fall into one of three main categories: memory-based, model-based, and hybrid. This work adopted memory-based CF, a method whose most

critical component is the mechanism of finding similarities between items and/or users (Su & Khoshgoftaar, 2009). The reason to choose CF is that computing new links between problems and solutions can be seen as introducing potential relevant elements from each other. CF is good at recommending potentially-related elements from one group to another. Many different methods exist to compute similarity (Ahn, 2008), and this work focused on log-likelihood, City Block and Tanimoto, all of which are detailed in (S. Guo et al., 2014) and they have proven effective in the experiments.

- Memory-based CF.

  Memory-based CF are based on full initial ratings in a user-item matrix (Ghazarian & Nematbakhsh, 2015), which is usually sparse (K. Yu et al., 2004). It's effective and easy to implement. Commonly used approaches in this technique are neighborhood-based CF and item-based or user-based top-N recommendations (Breese et al., 2013). For example, in user based CF algorithm, the value of ratings user u gives to item i is calculated as an aggregation of some similar users rating of the item.

  Given two points in the n-dimensional space A and B with the Cartesian coordinates of $(A_1, A_2, , A_i, A_{i+1}, , A_n) and (B_1, B_2, , B_i, B_{i+1}, , B_n)$, respectively. There are several measurements to calculate the similarities between two points, three of which are introduced in the following contents. These tree measurements are used for novel idea generation.

- City Block.

  City Block is also called Manhattan distance. It is aiming at searching for the shortest distance between two points in square city building blocks regardless of the one way street. Mathematically, it equals to the sum of distances of the segment AB(vector) projected to the axes.

- Tanimoto similarity.

Tanimoto similarity measurement is usually applied to boolean vectors, where each component is only 0 or 1. It indicates the ratio between the characteristics of both two vectors (the two vectors are being compared) and the union characteristics of both of them.

- log-likelihood similarity. Log-likelihood similarity is similar to Tanimoto coefficient similarity, it also measures the similarity for Boolean data. The difference from Tanimoto coefficient metric is that it emphasizes the unlikelihood of the two arrays (Owen et al., 2011).

2.3 Idea flow analysis

The scope of *Idea flow* in this study is limited to scientific paper citations. The unit that constitutes the *flow* is the single citation sentiment, which is represented by sentiment degree described in 3.2.5.

Unlike the concept of idea flow in industry (one of the concepts) [9], which sees *idea flow* as a process to shared brain for organizations and a platform for managing your networks of people, ideas, and knowledge, , idea flow in this work is focusing on analyzing the citation sentiment changes within or cross papers.

2.3.1 Word2vec

Mikolov *et al.* introduced word2vec technique (Mikolov, Chen, et al., 2013) that can obtain word vectors by training text corpus. The idea of word2vec (word embeddings) originated from the concept of distributed representation of words (M. J. Hinton Geoffrey & Rumelhart, 1986). The common method to derive the vectors is using neural probabilistic language model (Bengio et al., 2003). Word embeddings proved to be effective representations in the tasks of sentiment analysis (Tang et al., 2014; Xue, Fu, & Shaobin, 2014; Zhang et al., 2015) and text classification (Lilleberg et al., 2015). Sadeghian and Sharafat (Sadeghian &

[9] *https* : *//www.ideaflow.io/*

Sharafat, n.d.) extended word embeddings to sentence embeddings by averaging the word vectors in a sentiment review statement. Their results showed that word embeddings outperformed the bag-of-words model in sentiment classification.

When using word2vec, two parameters need to be specified, namely: the minimum word count and the context window. The context window determines which contextual neighbours are considered when estimating the vector representations. Another prominent hyper-parameter is the maximum window size (i.e. the maximum distance between the focus word and its contextual neighbours). Larger windows are known to induce embeddings that are more topical or associative, improving their performance on analogy test sets, while smaller windows induce more functional and synonymic models, leading to better performance on similarity test sets (Lison & Kutuzov, 2017).

2.3.2 Argumentative zoning

Topic modeling, text summarization and main idea identification are distinct NLP applications. They do share certain similarities. They all have the same primary purpose, which is to condense information. A topic could be represented by a sentence, a phrase or a discrete piece of content, expressing a specific subject. The main idea is the key of an article. Rather than using scattered words as used in the topic, main ideas are usually carried by a piece of words from the contents. The main idea is expected to maintain a clear meaning. In comparison, the scattered words in the topic may break the original meaning that the individual words carrying. Summarization is a passage or a write-up without changing its meaning but by using different words and sentences from the original contents (Blake, 1993). Summary or abstract does not only contain the main idea, but also includes sub ideas. In scientific document summarization, Argumentative Zoning (AZ) (Teufel, 2010) has been recognized as an effective approach, which aims at categorizing rhetorical sentences in publications. Researchers developed the task of AZ from different perspectives. In order to deal with the issue of lacking enough ground truth data, the paper (Y. Guo et al., 2011) proposed a

weakly-supervised approach that is based on the combination of active learning and selftraining, yielding a high accuracy of 81-percent when using only 10-percent of the labeled data. In the paper (Merity et al., 2009), the task of AZ was explored using Viterbi algorithm, which is effective for sequence tagging. They announced that their result is a 23-percent F-score increase on the Computational Linguistics conference papers marked up by Teufel (1999). AZ was originally restricted to one discipline, computational linguistics (CL). The paper (Teufel et al., 2009) is aimed at extending AZ to other domains. They used a domain expert to encode basic knowledge about the subject as part of the annotation guidelines. Their results indicated that non-expert human coders can use the guidelines to reliably annotate this scheme in two domains, chemistry and computational linguistic. To extract features from the sentences automatically is still challenging and needs more exploration.

Analyzing the rhetorical status of sentences manually requires huge amount of efforts, especially for structuring information from multiple documents. Fortunately, computer algorithms have been introduced to solve this problem. With the development of artificial intelligence, machine learning and computational linguistics, Natural Language Processing (NLP) has become a popular research area (Manning & Schütze, 1999; Hirschberg & Manning, 2015). NLP covers the applications from document retrieval, text categorization (Jackson & Moulinier, 2007), document summarization (Cao et al., 2015) to sentiment analysis (Nasukawa & Yi, 2003; Pang & Lee, 2008). Those applications are targeting different types of text resources, such as articles from social media (Asur et al., 2010) and scientific publications (Teufel & Moens, 2002). There are several approaches to tackle these tasks. From the machine learning prospective, text can be analysed via supervised (Teufel & Moens, 2002), semi-supervised (Sindhwani & Melville, 2008) and unsupervised (Jain & Farrokhnia, 1991) algorithms.

Document summarization from social media and news circles has been addressed from many angles, one of which is feature extraction and representation. At the early stage of document summarization, features are usually engineered manually.

Although the hand-crafted features have shown the ability for document summarization and sentiment analysis (Das & Martins, 2007; Pang & Lee, 2008), there are not enough efficient features to capture the semantic relations between words, phrases and sentences. Moreover, building a sufficient pool of features manually is difficult, because it requires expert knowledge and it is time-consuming. Teufel et. al. (Teufel & Moens, 2002) have built feature pool of sixteen types of features to classify sentences, such as the position of sentence, sentence length and tense. Widyantoro et. al. used content features, qualifying adjectives and meta-discourse features (Widyantoro et al., 2015) to explore AZ task. It took efforts to engineer these features and it is also time consuming to optimize the combination of the entire features. Therefore, computer assisted technologies such as document summarization are expected to play a role in condensing information and providing readers with more relevant short texts. Unlike document summarization from news circles, where the task is to identify centroid sentences (Radev et al., 2000) or to extract the first few sentences of the paragraphs (C.-Y. Lin & Hovy, 1997), summarization of scientific articles involves extra text processing stage (Teufel & Moens, 2002). After highest ranked texts are extracted, rhetorical status analysis will be conducted on the selected sentences. Rhetorical sentence classification, also known as argumentative zoning (AZ) (Teufel et al., 2000), is a process of assigning rhetorical status to the extracted sentences. The results of AZ provide readers with general discourse context from which the scientific ideas could be better linked, compared and analyzed. For example, given a specific task, which sentences should be shown to the reader is related to the features of the sentences. For the task of identifying a paper's unique contribution, sentences expressing research purpose should be retrieved with higher priority. For comparing ideas, statements of comparison with other works would be more useful. Teufel et. al. (Teufel & Moens, 2002) introduced their rhetorical annotation scheme which takes into account of the aspects of argumentation, metadiscourse and relatedness to other works. Their scheme resulted seven categories of rhetorical status and the categories are assigned to full sentences. The seven categories are aim, contrast, own, background, other, basis and textual. The relation between AZ and idea

mining is: the category of OWN potentially holds the novel ideas contained in the examined paper, whereas, the sentences with the category *aim* probably have the *research problem* contents in them. The results of AZ provide sentence level of resources for the experiments of idea flow, which is introduced in the following sections.

Examples [10] of human annotated sentences with their rhetorical status are shown in Table. 2.1.

TABLE 2.1: Examples of annotated sentences with their rhetorical status

Rhetorical Status	Examples
AIM	This paper discusses the lexicographical concept of lexical functions Mel'cuk and Zolkovsky 1984 and their potential exploitation in the development of a machine translation lexicon designed to handle collocations.
CTR	In two of the tasks, the training data is generated by a probabilistic context-free grammar and in both tasks our algorithm outperforms the other techniques.
OWN	We have explored examples of the kinds of tree sets and string languages that this system can generate.
BKG	English has a very limited system, marking little more than plurality on nouns and a restricted range of verb properties.
OTH	For this small example, writing such an apply predicate is not difficult.
BAS	Following Pereira et al. 1993, we measure word similarity by the relative entropy, or Kullback-Leibler distance, between the corresponding conditional distributions.
TXT	The next section describes the binary representation and the length formul derived from it in detail; readers satisfied with the intuitive descriptions presented so far should skip ahead to the Phonotactics sub-section.

2.3.3 Sentiment analysis

Broadly speaking, sentiment analysis (Mntyl et al., 2018) or opinion mining is to compute the opinions, attitudes or emotions toward entities, as well as their attributes (Liu & Zhang, 2012).

The evolution of scientific ideas happens when old ideas are replaced by new ones. Researchers usually conduct scientific experiments based on the previous publications. They either take use of others work as a solution to solve their specific problem, or they improve the results documented in the previous publications by

[10] These texts were randomly selected from Argumentative Zoning Corpus, which is described in dataset section.

Citing	Cited	Polarity	Examples
A1	A0	Positive	One of the most effective taggers based on a pure HMM is that developed at Xerox (Cutting *et al.* , 1992).
A2	A0	Negative	Brill's results demonstrate that this approach can outperform the Hidden Markov Model approaches that are frequently used for part-of-speech tagging (Jelinek, 1985; Church, 1988; DeRose, 1988; Cutting *et al.* , 1992; Weischedel *et al.*, 1993), as well as showing promise for other applications.

TABLE 2.2: Examples of positive and negative citations.

introducing new solutions. Citation sentence examples[11] with different sentiment polarity are shown in Table 2.2.

Sentiment analysis of citations plays an important role in plotting scientific idea flow. We can see from Table 2.2, one of the ideas introduced in paper A0 is *Hidden Markov Model (HMM) based part-of-speech (POS) tagging*, which has been referenced positively in paper A1. In paper A2, however, a better approach was brought up making the idea (*HMM based POS*) in paper A0 negative. This citation sentiment analysis could lead to future-works in such a way that new approaches (mentioned in paper A2) are recommended to other papers which cited A0 positively[12]. Analyzing citation sentences during literature review is time consuming. Recently, researchers developed algorithms to automatically analyze citation sentiment. For example, (Abu-Jbara et al., 2013) extracted several features for citation purpose and polarity classification, such as reference count, contrary expression and dependency relations. Jochim *et al.* tried to improve the result by using unigram and bigram features (Jochim & Schütze, 2014). (Athar, 2011) used word level features, contextual polarity features, and sentence structure based features to detect sentiment citations. Although they generated good results using the combination of features, it required a lot of engineering work and big amount of annotated data to obtain the features. Further more, capturing accurate features relies on other NLP techniques, such as part-of-speech tagging (POS) and sentence parsing. Therefore, it is necessary to explore other techniques that are free

[11] `Randomly selected from : http://cl.awaisathar.com/citation-sentiment-corpus/`
[12] Restriction: the citations share the similar topics. In this case: *HMM based POS tagging*

from hand-crafted features. With the development of neural networks and deep learning, it is possible to learn the representations of concepts from unlabeled text corpus automatically. These representations can be treated as concept features for classification. An important advance in this area is the development of the word2vec technique (Mikolov, Chen, et al., 2013), which has proved to be an effective approach in Twitter sentiment classification (Tang et al., 2014).

According to my intuition, the relation between citation sentiment analysis and idea detection is that ideas are surrounded by sentiment. The reactions from researchers who captured the citation sentiment changes can lead to novel idea creation. To conduct citation sentiment analysis, supervised learning with automatic extracted features can be a good choice.

2.3.4 Texture analysis in image processing

Image texture quantifies the information of the spatial arrangement of color or intensities in an image. Researchers have developed algorithms to extract texture features. Gray Level Co-occurrence Matrix (GLCM) has proved to be a useful statistical method for extracting textural features from images.

GLCM has been defined as a positive NN matrix (Haralick et al., 1973), where N is the number of gray levels in an image. The elements in the matrix are generated by counting the number of occurrences of pixel pairs (with gray levels i and j) given a certain distance and direction. The minimum distance can be set to one pixel and the maximum distance can be set to the half of the image size.

Using GLCM to measure the texture features of an image leads to the idea of developing a similar algorithm to measure the *texture feature* of citation sentiment across a paper. Here, *texture feature* of citation sentiment within a paper refers to the citation sentiment changes between positive and negative.

2.3.5 Sentiment Level Co-occurrence Matrix (SLCM)

As introduced above, texture is used for measuring the spatial arrangement of color or intensities in an image or selected region of an image (Shapiro & Stockman, n.d.). The coarseness level can be classified by the texture features. It is reasonable to associate the coarseness level in an image with the controversial degree in the sequence. Here, coarseness refers to the entire opinions (measured by the quality of them being rough or harsh) toward one cited paper in one citing paper. Based on this assumption, a measurement called Sentiment Level Co-occurrence Matrix (SLCM) for analyzing the sentiment changes in the citation sequences was proposed. An example of lower level coarseness image is that: an image with the half part (say, the top of the image) totally black and the other half is totally white. If the gray scales in the image are flattened to one single vector, it would be like this: $[0, 0, ..., 255, 255]$ (there is no 255 in between the 0s, and likewise), which is with lower coarseness level. In comparison, an image with zebra pattern is more coarse. The zebra-like pattern can be expressed with the vector like this $[0, 0, 255, 0, 0, 255, 255, 0, 0, 0, 0, 255, 0, 0, ..., 255, 0, 0, 255]$. Similarly, the citation sentiment toward one paper may have different patterns, which can be measured according to their coarseness. The purpose to classify the entire citation sentiment in one citing paper (towards one specific cited paper) is that, if it is with higher coarseness level, it may indicate that the cited paper is controversial, thus, there are more space to explore (research gap maybe detected). If it is almost all positive, that means the idea in the cited paper is dominating the research field (the citing paper is focusing) at that moment. If it is almost all negative, it maybe because the cited paper has a lot of flaws that need to be further researched.

There are several approaches to measure coarseness (texture). The reason to choose the one similar to GLCM [13] is that GLCM fits one dimensional vectors. Citation sentiment sequences can be seen as one dimensional vectors. Other concepts such as local binary pattern (LBP) (Ojala et al., 2002; Ahonen et al., 2006)

[13]Note that in this experiment, there are no applications based on the statistics *contrast*, *correlation*, *energy* or *homogeneity* as derived from GLCM.

may also be considered as an inspiration for developing algorithm to measure citation sentiment changes.

To measure the sentiment changes, four simple prototypes to detect the internal sentiment change were defined. Figure 2.5 demonstrates the prototype patterns. All of the prototypes contain two units, representing the adjacent two citations (the neutral citations were ignored, therefore, there are only two levels of sentiment: 1 and -1, representing positive and negative respectively). *Adjacent* means the two citations appeared next to each other on the citing paper. The pair of rectangles reflect the sentiment change: the white-white (black-black) pattern stands for there is no sentiment change and both of the adjacent citations are positive (negative); the white-black pattern represents the sentiment change from positive to negative and the black-white pattern means the sentiment changes from negative to positive[14]. Similar to generating local binary pattern features (Mäenpää, 2003), a vector containing four elements is constructed from the SLCM, which will be used as the sentiment feature to classify the citation sentiment sequence. The four prototypes will run through the sequence one citation at a time and the occurrences of each prototype will be saved to the SLCM.

2.3.6 Eddy Dissipation Rate

The Eddy Dissipation Rate (EDR) is a metric for representing the intensity of atmospheric turbulence (AIAA, 2012) and has been used for turbulence verification (Takacs et al., 2005).

According to Kolomogrov (Kaneda et al., 2003) hypothesis, the large eddies in the atmosphere are inherently unstable. These large eddies break up and transfer their energy to smaller size eddies. This transfer of energy continues from larger to smaller eddies, until viscous dissipation becomes dominant and the turbulence kinetic energy (TKE). TKE is converted into heat. The rate at which the TKE is dissipated is known as the EDR. That is to say, EDR is the rate at which the

[14]For GCSS, this texture feature analysis was only applied on the Measurement G-1: $countcitations_g$

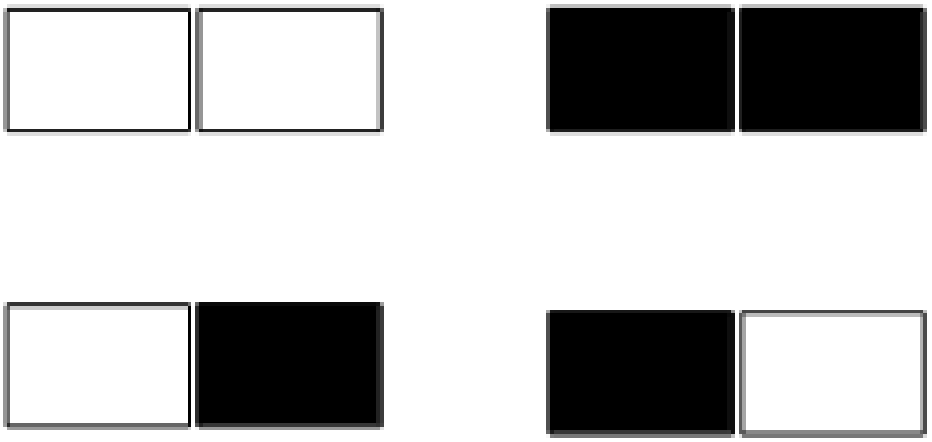


FIGURE 2.5: Feature prototypes for detecting sentiment change. All of the prototypes contain two units, representing the adjacent two citations. The white-white (black-black) pattern stands for there is no sentiment change and both of the adjacent citations are positive (negative); the white-black pattern represents the sentiment change from positive to negative and the black-white pattern means the sentiment changes from negative to positive.

TKE is absorbed by breaking down eddies into smaller and smaller eddies until it is converted into heat by viscous forces.

The reason to use EDR to measure the idea flow is that it is hypothesized that the citation sentiment changes could be smooth or (strongly) turbulent. By smooth, I mean, if one citation got one positive cite during one year and got a negative cite next year, then, the attitude towards the idea in the cited paper is considered to be smooth. In comparison, if one paper got a 100 positive cites during one year and got 1000 negative cites next year, then, the readers attitude towards the idea in that paper is not smooth.

Chapter 3

Research Methodology

This chapter covers dataset review (3.1) and methodology explanation (3.2).

3.1 Dataset review

3.1.1 Dataset for idea mining and novel idea computation

- Description

  A collection of the titles and abstracts were extracted randomly from 3,665 English language articles published in the journal *Science*. The dataset, covering the years 1998-2015, was partitioned so that half of the articles formed the training set and the other half of the test set. From the entire dataset, noun-phrases were obtained using part-of-speech-tagging technique. After noun-phrase filtering, the raw noun-phrases pool contained 57,621 noun-phrases and noun-phrase categorization resulted in 54,073 p-phrases and 3,548 s-phrases.

- Purpose

  This dataset is used for mining ideas using noun-phrases detection and classification as well as novel idea computation.

- Dataset Samples

TABLE 3.1: Examples of title and abstract from journal Science.

(published year) title and abstract
(1998) Conservation of T Cell Receptor Conformation in Epidermal Cells with Disrupted Primary V Gene Usage. A feature that distinguishes T cell subsets from most T cells and B cells is the association of expression of single T cell receptor (TCR) and variable (V) region gene segments with specific anatomic sites. Mice lacking the TCR V5 chain normally expressed by most dendritic epidermal T cells were shown to retain a conformational determinant (idiotype) ordinarily expressed exclusively by such V5+cells. Conservation by shuffled TCR chains of an idiotype associated with a specific anatomic site indicates that for TCR, as for immunoglobulin, conformation is associated to a greater extent with the function or development of lymphocyte repertoires than is the use of particular gene segments.
(2010) Sex-Specific Parent-of-Origin Allelic Expression in the Mouse Brain. Genomic imprinting results in preferential gene expression from paternally versus maternally inherited chromosomes. We used a genome-wide approach to uncover sex-specific parent-of-origin allelic effects in the adult mouse brain. Our study identified preferential selection of the maternally inherited X chromosome in glutamatergic neurons of the female cortex. Moreover, analysis of the cortex and hypothalamus identified 347 autosomal genes with sex-specific imprinting features. In the hypothalamus, sex-specific imprinted genes were mostly found in females, which suggests parental influence over the hypothalamic function of daughters. We show that interleukin-18, a gene linked to diseases with sex-specific prevalence, is subject to complex, regional, and sex-specific parental effects in the brain. Parent-of-origin effects thus provide new avenues for investigation of sexual dimorphism in brain function and disease.

3.1.2 Dataset for detecting ideas using neural networks

- Description

  Four hundred and fifty abstracts were obtained and analysed from the Proceedings of the Neural Information Processing Systems conference (NIPS). The publication year: 2008 - 2016; Around 50 abstracts per year. The reason to choose NIPS as the data source is that NIPS have more technical papers, which contains problem and solution stated clearly.

  The entire 350 annotated abstracts from NIPS were randomly split into 3 parts: 150 (train), 100 (development) and 100 (test). The development is used for tuning the parameters or perform other actions to make the model better, such as early stopping. All the words were converted to lower case and punctuations were removed. The vocabulary size was 4046, with 1014 as the training set, 722 as the development set and 711 sentences as the test set.

- Purpose

The purpose of this work is to obtain a problem solution sequences (PSS) that represent the most important problem and solution in the abstract, thus, it enables machine learning techniques on idea detection.

- Annotation Scheme

  Seven annotators are the researchers from Computer Science, who manually highlighted PSS on printed abstracts with coloured highlighter pens. The guiding principle underlying the annotation scheme is to keep the sequence as short as possible, while retaining enough information to distinguish the novel contribution of the paper.

  The annotation task was conducted using the title and the abstract of the corpus of papers. The annotation rules were as follows:

  A Problem (Solution) sequence might be a word, a phrase or an entire sentence. However, the sequence should not be separated by other words.[1]

  For each abstract, it is expected to identify only one problem and one solution. If there is more than one problem (or solution) in the abstract, the one that is most important should be chosen. When there are multiple PSS in a paper, then the main problem (or solution) should be those that are most related to the title. The relatedness is determined by the annotators intuition.

  The chosen sequence should reflect the novelty of the paper.

  The chosen sequence should be as short as possible.

  The distance between the chosen sequence and the root of the sentence should be as close as possible.[2]

  If there is no problem or solution phrase identified, the abstract is labeled as non-PSS abstracts, and excluded from the analysis.

- Data sample

  An example of the human annotated PSS is shown in the Figure 3.1.

[1]The reason behind this restrictions is, if the selected phrases are separated by other words, it would be difficult to infer the semantic relations between the scattered concepts.

[2]The distance could be measured by the depth level on the parsed dependency tree.

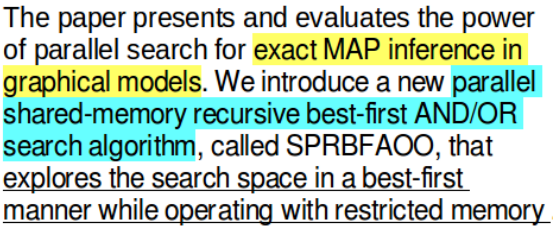


FIGURE 3.1: An example of human annotated abstract. The *problem* (*solution*) sequences were highlighted in yellow (blue). The sequence underlined is the solution candidate that was discarded by the annotator.

3.1.3 Dataset for Argumentative Zoning (AZ)

- Description

  There are two parts of datasets being used in this task. One part is used for training word2vec and the other one is for evaluation.

  For training word2vec, ACL Anthology Reference Corpus (*ACL* collection) [3] and *MixedAbs* collection were collected.

  ACL collection contains the canonical 10,921 computational linguistics papers, from which 622,144 sentences were generated after filtering out sentences with lower quality (e.g.: the sentences containing less than 3 words, which might be some fragments from citations.).

  MixedAbs collection contains 6,778 sentences, extracted from the titles and abstracts of publications provided by WEB OF SCIENCE [4].

  To test the word2vec technique for the task of AZ, Argumentative Zoning Corpus (*AZ* corpus) was adopted. *AZ* corpus consists of 80 AZ−annotated conference articles in computational linguistics, originally drawn from the Cmplg arXiv. [5]. After Concatenating sub-sentences, 7,347 labeled sentences were obtained.

- Purpose

  The purpose of these datasets were used for testing word2vec ability on classifying sentences from scientific papers into seven categories, namely:

[3] *http://acl-arc.comp.nus.edu.sg/*
[4] *webofknowledge.com*
[5] *http://www.cl.cam.ac.uk/~sht25/AZ_corpus.html*

BKG: General scientific background

OTH: Neutral descriptions of other people's work

OWN: Neutral descriptions of the own, new work

AIM: Statements of the particular aim of the current paper

TXT: Statements of textual organization of the current paper

CTR: Contrastive or comparative statements about other work; explicit mention of weaknesses of other work

BAS: Statements that own work is based on other work (purple)

- Data Samples

Table 3.2: Examples of ACL Anthology and AZ Corpus.

(dataset) randomly selected sentences
(ACL Anthology) Carnegie Group Inc. participated in both the dry run and SUMMAC evaluations. CGI's performance was encouraging and relied not only on the MMR metric but on traditional sentence selection methods, such as, always select the first sentence of a document. The MMR metric has proven to be extremely flexible and has application in many facets of retrieval and summarization. MMR is expected to prove most useful in the summarization of multiple documents. (ACL Anthology) 2.1. VARYING VIEWPOINT ON THE OBJECT. Mainly this involves the contrast between a close-up and a remote view of the objects. Most often, this is not a matter of actual distance, but a way of viewing an object for a given purpose: one may choose to ignore one or more dimensions, or internal characteristics of the object. For example, a road may be seen as a strip
(AZ Corpus) $< A-SID =' A-0'AZ =' BKG' >$ The temporal relations that hold between events described by successive utterances are often left implicit or underspecified . $< /A-S >$ $< A-SID =' A-1'DOCUMENTC =' S-73'AZ =' AIM'TYPE =' ITEM' >$ We address the role of two phenomena with respect to the recovery of these relations : $< /A-S >$ (AZ Corpus) $< SID =' S-0'AZ =' BKG' >$ Abstract generation is , like Machine Translation , one of the ultimate goal of Natural Language Processing . $< /S >< SID =' S-1'AZ =' CTR' >$ However , since conventional word-frequency-based abstract generation systems ¡REF TYPE='P'¿Kuhn 1958¡/REF¿ are lacking in inter-sentential or discourse-structural analysis , they are liable to generate incoherent abstracts . $< /S >$

- Characteristics

The characteristics of word embeddings based on different model and dataset are listed in Table. 3.3.

Table 3.3: Characteristics of word embeddings based on different model and dataset

	Number of features	Vocabulary size
AVGWVEC ACL+AZ 300	300	13685
AVGWVEC ACL+AZ 100	100	14325
PARAVEC ACL+AZ 100	100	74261
AVGWVEC MixedAbs 100	100	4126
AVGWVEC BSWE 100	100	643
AVGWVEC Brown model	100	56057

- Credit

  The *ACL* collection was credited to ACL Anthology. *AZ* corpus was credited to Professor Simone Teufel and her team [6].

3.1.4 Dataset for citation sentiment analysis

- Description

  The first dataset ($dataset-basic$) was originally taken from ACL Anthology (Bird et al., 2008). Athar and Awais (Athar, 2011) manually annotated 8,736 citations from 310 publications in the ACL Anthology. All of the labeled sentences (830 positive, 280 negative and 7,626 neutral) were used for testing. [7]

  The second dataset ($dataset-implicit$) was used for evaluating implicit citation classification, containing 200,222 excluded (x), 282 positive (pos), 419 negative (neg) and 2,880 neutral (neu) annotated sentences. Every sentence which does not contain any direct or indirect mention of the citation is labeled as being excluded (x)[8]. The third dataset (dataset-pn) is a subset of dataset-basic, containing 828 positive and 280 negative citations. Dataset-pn was used for the purposes of (1) evaluating binary classification (positive versus negative) performance using sent2vec; (2) Comparing the sentiment classification ability of PS-Embeddings with other embeddings.

- Purpose

  To evaluate the word2vec performance on citation sentiment classification.

- Dataset Samples

[6] $https://www.cl.cam.ac.uk/\ sht25/az.html$

[7] In (Athar, 2011)'s work, they used 244 negative, 743 positive and 6277 neutral citations for testing.

[8] $http://www.cl.cam.ac.uk/\tilde{}aa496/citation\text{-}context\text{-}corpus/$

TABLE 3.4: Examples of $dataset - basic$ and $dataset - implicit$.

(dataset) randomly selected sentences
($dataset-basic$) Unlike well-known bootstrapping approaches (Yarowsky, 1995), EM and CE have the possible advantage of maintaining posteriors over hidden labels (or structure) throughout learning; bootstrapping either chooses, for each example, a single label, or remains completely agnostic.
($dataset-implicit$) Our model exploits the same kind of tag-n-gram information that forms the core of many successful tagging models, for example, (Kupiec, 1992), (Merialdo, 1994), (Ratnaparkhi, 1996).

3.1.5 Dataset for idea flow analysis

- Description

  In the paper (Athar & Teufel, 2012a), the author annotated 20 papers. Based on these 20 annotated papers, the following two sub datasets were created: $Data_I$ and $Data_G$, representing internal and global citation respectively.

  The $Data_I$ contains 1022 < cited-paper, citation-sentence > pairs, among which, 110 were annotated (24 positive, 54 negative and 32 controversial) for the ICSS analysis[9].

  The $Data_G$ contains 20 < cited-paper, citation-sequence > pairs (1 positive, 7 negative and 12 controversial), where were derived from 3581 < cited-paper, citation-sentence > pairs (282 positive, 419 negative and 2880 neutral). The value in the $citation - sequence$ was obtained by analyzing the citing paper's citations, which was explained in the section 3.2.5.

- Purpose

  The dataset $Data_I$ was used for analyzing idea flows internally whereas $Data_G$ was used for analyzing idea flows globally.

- Human annotation to determine the controversy and polarity

  Before assigning a sentiment label to each of the citations, a general classification was performed to group all the citations into $has_{sentimentchange}$ and $no_{sentimentchange}$, based on the hard threshold: both the number of positive and negative citations are more than 1. Fine tuned classification was then conducted on the group $has_{sentimentchange}$. The rules to determine the citation sentiment type are described as follows[10]:

[9] as explained early, the citation sequences that have no sentiment change were ignored

[10] This is subjective, more research is needed to tell where the threshold is.

$$sg_i = \begin{cases} 1, & \text{1 dominates the sequence} \\ -1, & \text{-1 dominates the sequence} \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

where domination is determined by crowd-sourcing. Reader helped to assign a single label to each sequence either with 1, -1 or 0. The label that has exceeding 60 percents votes from all the participants is considered dominating. In the end, 110 sequences were annotated.

- Dataset Samples

TABLE 3.5: Samples of $Data_I$

sequence	polarity
0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 1 0 1 0 1 1	1
1 -1 -1 -1	-1
0 0 0 -1 -1 -1 0 0 0 0	-1
1 1 -1 0 0 0 0 0 0 0 0 0 0 1	1

TABLE 3.6: Samples of citation sequence with raw citation counts

cited paper (citing pape)	citation sequence using raw counts
P07-1033 (P08-1029)	0 -1 -1 0
A92-1018 (A94-1009)	0 0 0 1 1 1 1 -1 -1 1 0 3
W06-1615 (D07-1129)	0 0 0 0 0 0 0 0 0 -1 -1 -2
J93-1007 (P97-1061)	0 -1 0 0 0 0 -1 -2

TABLE 3.7: Samples of $Dataset_G$

PaperID	citation sequence using the proportion of positive citation counts among the total number of citations
J93-1007	0.0 0.0 1.0 0.5 0.0 0.3 0.0 0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.3 0.0 1.0 0.5 0.2 0.0 0.0 1.0 0.0
J90-1003	0.5 0.0 0.3 0.5 0.0 1.0 0.0 0.5 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 1.0 0.5 0.0 0.0 0.0 0.3 0.0 0.4 1.0
W05-0909	0.0 1.0 0.2 0.0 0.0 1.0 0.4 0.5 1.0 1.0 0.7 1.0 1.0 1.0 0.5 0.0 0.25 1.0 1.0 0.0 0.4 1.0 0.0 1.0 0.7
W04-1013	1.0 1.0 0.0 1.0 0.6 1.0 1.0 1.0 1.0 0.2 1.0 1.0 0.6 1.0 0.6 1.0 0.0 1.0 0.0 1.0 0.0 0.7 1.0 1.0 0.0

3.2 Methodology

3.2.1 Idea mining and novel idea computation

Idea mining.

Young *et al.* (Young, 1975) describe two principles for producing novel ideas:

- An idea is nothing more or less than a new combination of old elements.
- The capacity to bring old elements into new combinations depends largely on the ability to see relationships.

Based on these principles, it is reasonable to argue that novel idea candidates can be established by uncovering the relationships between *problems* and *solutions* within scientific texts. These components can then be intelligently recombined into previously unforeseen <*problem*,*solution*> pairs ready for consideration by researchers.

This constructive definition of an "idea" echoes Thorleuchter *et al*'s use of the term, who themselves reference the definitions in (Wallas, 1926) in their attempts to identify concepts within various text corpora. In (Thorleuchter, Van den Poel,

& Prinzie, 2010b) they define an idea as a combination of a *means* and an appertaining *ends*, using unconstrained term vectors to represent each of these entities. In contrast, the representations of *problems* and *solutions* are using noun-phrases. This assumption is based on previous studies (Hildreth & Kimble, 2004; Kamp, 1981) which indicate that while a sentence's main conceptual information is usually expressed by both noun and verb phrases. [11].

Defining an "Idea".

Considering a document T, represented as an ordered set of N words, where $T = \langle w_1, w_2, \ldots, w_N \rangle$, then the functional definitions used to construct the representation of an idea are as follows:

Noun-phrase: A *noun-phrase*, ϕ, is an ordered subset of the text, extracted from T (in this case extracted from the titles or abstracts of publications using part-of-speech tagging technique):

$$\phi = \langle w_1, w_2, \ldots, w_n \rangle. \tag{3.2}$$

P-phrase: A *p-phrase* is defined as a noun-phrase determined to be a *scientific problem*. $\mathcal{P}_T$ is defined as the set of m p-phrases extracted from a document, T (where $m \leq N$):

$$\mathcal{P}_T = \left\{ \phi_{(p,1)}, \phi_{(p,2)}, \phi_{(p,3)}, \ldots \right\} \tag{3.3}$$

S-phrase: An *s-phrase* is defined as a noun-phrase that has been categorized as a *technical solution* or a *methodological approach*. $\mathcal{S}_T$ that represents the set of q p-phrases extracted from the document, T (where $q \leq N$):

$$\mathcal{S}_T = \left\{ \phi_{(s,1)}, \phi_{(s,2)}, \phi_{(s,3)}, \ldots \right\}, \textit{where } \mathcal{S}_T \cap \mathcal{P}_T = \emptyset \tag{3.4}$$

[11] For the first experiment to detect ideas and generate novel ideas, only noun phrases are taken into account. Because if including verbs, the verbs should be extracted together with the nouns that appear right after them. In that case, the less occurrences of the verb-noun would lead to sparse matrix in recommender system, which may fail to generate novel ideas.

Idea: A specific *idea*[12] can then be defined as a combination of some p-phrase and s-phrase extracted from dataset, D:

$$idea = \langle p\text{–}phrase, s\text{–}phrase \rangle \tag{3.5}$$

Known Idea: A *known-idea* is defined as a combination of any p-phrase and s-phrase that are found in the same document, T:

$$known\text{-}idea = \{\exists T \; known\text{–}idea \in \mathcal{P}_T \times \mathcal{S}_T\} \tag{3.6}$$

Known-ideas may additionally be attributed a relevance value, representing some measure of the idea's significance within the literature. In this work, four types of statistics were used to measure this significance, described in more detail in 3.2.1.

Novel Idea: A *novel-idea* is the combination of some p-phrase and s-phrase from the dataset, but which do not co-occur in the same document:

$$novel\text{-}idea = \{\exists T \exists U \; (novel\text{–}idea \in \mathcal{P}_T \times \mathcal{S}_U) \wedge (T \neq U)\} \tag{3.7}$$

Novel-ideas may also be assigned a value that reflects the strength of the relationship between its *p-phrase* and *s-phrase* components (as discussed in 2.2).

The method proposed in this work focuses on discovering problem-solution relationships between noun-phrases as detected in the abstracts (together with their titles) of scientific papers. An abstract is a fully self-contained, capsule description of a paper (Koopman, 1997). The noun-phrases it contains should reflect the issue(s) that the author(s) wish to address, so a list of noun-phrases extracted from it provides an ideal foundation for the seed pool of *p-phrases*. If an s-phrase

[12] I do not claim that <problem, solution> pairs represent a universal definition of ideas, but are a useful entity pragmatically (reflecting a related construct that is readily amenable to computation).

is detected in the same piece of text, semantic relationships between it and neighbouring p-phrases are established[13]. Based on this premise, the approach to subsequently computing novel idea candidates can be broken down into six stages:

1. Noun-phrase extraction from a training-set corpora using Part-Of-Speech tagging.
2. Phrase filtering to remove stop words and text with low information content.
3. Classification of noun-phrases into *p-phrases* (problems) and *s-phrases* (solutions).
4. Aggregation of highly co-occurring <*p-phrase*,*s-phrase*, *relevance*> known-idea triples.
5. Processing of this set of known-idea triples via a collaborative filtering (refer to 2.2.2) mechanism.
6. Assessment of the resulting ranked list of novel idea candidates that is output.

Several of the steps in this automated process analogise to specific stages in traditional brainstorming sessions. This is demonstrated in Fig. 3.2, which shows the process of the novel idea computation system on the right, and the corresponding steps in real-world ideation sessions on the left. Each of the stages were examined in more detail below.

Noun-phrase extraction. The first step in this method involves the detection of noun-phrases within the titles and abstracts of the publications that make up the training set. This is undertaken using a standard Part-of-speech (POS) tagging algorithm[14]. While there exist more complex linguistic indicators of an "idea", there are numerous advantages in assuming that noun-phrases are sufficient to represent the informational content of concepts: they are computationally parsimonious; their detection is well understood algorithmically; and studies show that such n-grams preserve far more semantic content than individual term extraction (Ding

[13]For example, consider the sentence *"a dynamic panel data estimation technique is used to examine the effects of internal demand on domestic credit"*. The n-gram "dynamic panel data estimation technique" will be recognised as an *s-phrase*, and associated with co-occurring *p-phrases* such as "effects of internal demand on domestic credit").

[14]In this study the *CiteSpace* application for POS tagging (C. Chen, 2006) was used, which was found performed better than other options such as the *TextBlob* Python library.

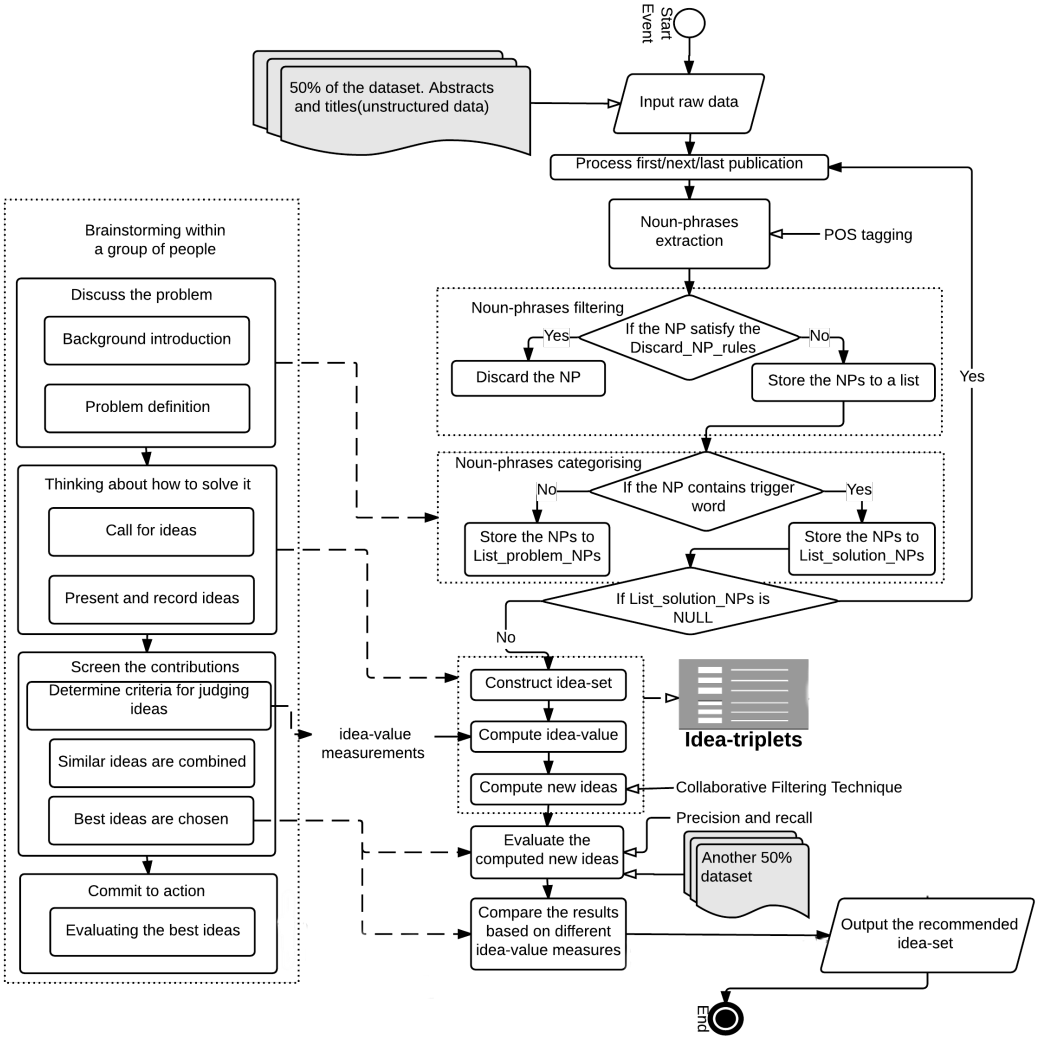


FIGURE 3.2: System flow - each of the method's processing steps are listed on the right hand side, with corresponding stages in real-life brainstorming sessions mapped on the left.

& Chen, 2014). Recalling the definition of a noun-phrase, $\phi = \langle w_1, w_2, \ldots, w_n \rangle$, for each document, T, a list of noun-phrases: $\Phi_T = \{\phi_1, \phi_2, ...\}$ were produced.

Noun-phrase filtering. Some noun-phrases generated by the POS tagger are not suitable for inclusion within idea construction. In particular, some concepts will be semantically redundant (i.e. they will have minimal information content in the same vein as *stop words* in traditional informational retrieval tasks). Examples in scientific abstracts are n-grams such as "I present a ***novel model***..." or "the ***general approach*** is tested through an evaluation procedure that...". This stage aims to eliminate such phrases, thus streamlining the method's subsequent

processing steps. To this end, two filtering steps were employed. First, given a set of hand-crafted "danger" terms W,bi-grams were removed that feature any of its elements:

$$\Phi_{filtered} = \{\phi \in \Phi_T : |\phi| = 2 \rightarrow \phi \cap \mathcal{W} = \varnothing\} \tag{3.8}$$

The equation 3.8 indicates that if the phrases have 2 words and one of them is in the "danger" set, they will not be excluded.

The version of $\mathcal{W}$ used in this experiments is listed in Table 3.8. In addition, a threshold was enforced on the frequency of retained noun-phrases. For this Jenks natural breaks classification method (Jenks, 1967) was used, assigning p-phrases into five categories according to their frequency across the corpus, eliminating all noun-phrases in the most frequent category. Jenks natural breaks is a data clustering method designed to determine the best arrangement of values into different classes. This is done by seeking to minimize average deviation of each class from the class mean, while maximizing deviation of each class from the means of the other groups. In other words, the method seeks to reduce the variance within classes and maximize the variance between classes. This step is based on the assumption that phrases that exhibit extremely high frequency either have low information content, or reflect noun-phrases that offer no untapped research value.

TABLE 3.8: Noun-Phrase filtering terms

overall, primary, key, valuable, excellent, potential, essential, unique, numerous, important, prior, practical, basic, different, simple, successful,
current, possible, previous, existing, well-established, independent, particular, usual, new, old, powerful, main, common, detailed, efficient, good, acceptable,
effective, novel, state-of-the-art, useful, modern, unreliable, additional,
methodological, available, recent, general, specific, creative, brief, critical, major, second, reasonable, various, personal, latest , interesting

Jenks natural breaks classification method, is a data clustering method for better arranging values into different classes. This method is achieved by seeking to minimize each class's average deviation from the class mean, while maximizing each class's deviation from the means of the other groups. That is to say, this

method is aiming at reducing the variance within classes and maximize the variance between classes.

Noun-phrases categorization. Now a filtered set of noun-phrases was obtained and they need to be categorized into two groups: *p-phrases* and *s-phrases* (representing problems and solutions respectively). There are numerous possible approaches to achieve this task, ranging from *named entity recognition techniques* (Nadeau & Sekine, 2007) to the application of *linguistic structure matching* (Bybee & Hopper, 2001). Unfortunately, in order to utilise these techniques a vast amount of annotated data is required, data that is as-yet-unavailable. Filtered noun-phrases, $\Phi_{filtered}$, are stemmed using the *porter stemming* algorithm (Willett, 2006), producing $\Phi_{stemmed}$. The compositions of phrases in this set are then compared with a compact bag of trigger words (shown in 3.9), $\mathcal{G}$, in order to explicitly identify *s-phrases*. Porter Stemming algorithm (Ali & Ibrahim, 2012) is a method to remove suffixes from words in English. This Stemmer is based on the idea that the suffixes in the English language (approximately 1200) which consists mainly of a combination of smaller and simpler suffixes. This Stemmer is a linear step Stemmer. In information retrieval tasks, removing suffixes automatically is very useful.

TABLE 3.9: S-phrase cue terms. The term "method" was used as a seed, with the set of trigger words being expanded through a combination of synonym extraction via *www.thesaurus.com* and isolating nearest neighbours using *Word2vec* (see https://code.google.com/p/word2vec/).

approach, technique, scheme, algorithm, analysis,modelling,methodology, strategy, framework, tool, procedure, structure, processing, heuristic, mechanism, architecture, theory, paradigm, formalism, platform

Examples of these cue terms contained in $\mathcal{G}$ might be "method", "approach" and "theory" (the set of trigger words used in the experiments and method of derivation is shown in Table 3.9). The method to choose those words as the solution cue words is using wordnet to generate the synonyms of the seed word "solution". Those noun-phrases which remain unmatched are subsequently designated as *p-phrases*[15]. The result is that for a document, T, a set of *s-phrases*, $\mathcal{S}_T$, and

[15]In some ways this is an algorithmic rendition of the arguable expression: "if you are not part of the solution, you are part of the problem".

p-phrases, $\mathcal{P}_T$ were produced [16], where:

$$\mathcal{S}_T = \{\phi \in \Phi_{stemmed} : \exists w \in \mathcal{G}\,[w \in \phi]\} \tag{3.9}$$

$$\mathcal{P}_T = \Phi_{stemmed} - \mathcal{S}_\mathcal{T} \tag{3.10}$$

This stage of classification has analogies to the real-world brainstorming processes discussed in 2.2.1 - in the "discuss the problem" stage of the process (Osborn, 1953), if no specific problem angle is specified, participants are instructed to conjure up any noun-phrases that are parts of the problem (i.e. people, places, entities, etc.) in a free-form fashion.

Known-idea construction. The algorithm must now enumerate *known-ideas* before it will be able to generate novel idea candidates through their recombination. It does this by pairing *p-phrases* and *s-phrases* deemed to be associated with each other. This pairing is done by permutation. In linguistic processing the specific relation types that are extant between noun-phrases can be uncovered using a range of extraction techniques such as kernel methods; dependency trees (Culotta & Sorensen, 2004); text pattern or structure creation (Thorleuchter, 2008); semantic graphs, topic templates and ontologies (e.g. WordNet) (Banko et al., 2008; Trampuš & Mladenic, 2014). However, due to the general qualities of a good abstract (Andrade, 2011) - i.e. it should be a condensed and concentrated version of the full text of the research manuscript - It is reasonable to assume that the concepts introduced in a single abstract are all related with each other regarding a specific topic domain. This assumption means it is possible to postulate valid idea-pairs simply by observing the co-occurrence of a problem and solution with the same abstract. It is noted that this approach may generate some unexpected pairings, such as $< yieldingtransporttimes, diapiricupwellingmodel >$ - this, however, is still in line with the general rules of brainstorming, where the pairing non-obvious components can expand the creativity of a real-world ideation session. The corresponding expansion of the idea pool can increase the chances of

[16] It indicates that if the phrase is not a solution, then, it belongs to a problem

producing a radical and effective solution, and as such, currently some traditional linguistic processing constraints were neglected:

1. the details of relationship types between noun-phrases were not integrated.
2. nor distances between the *root* and other nodes in the *Parse Tree*.

Once co-occurring <*p-phrase*,*s-phrase*> pairs have been identified they are assigned a score reflecting their "interestingness" or relevance to the corpus. This value, v, is necessarily subjective, and as such several approaches to determining the value was examined, as described in more detail in 3.2.1. Once a value is selected, the result of the idea construction process is the set of known-ideas, $\mathcal{K} = \{idea_1, idea_2, ...\}$, as summarised by algorithm 1 below.

Algorithm 1

procedure EXTRACT_KNOWN_IDEAS($\mathcal{D}$) ▷ $\mathcal{D}$ is the full document set
  $\mathcal{K} \leftarrow \emptyset$ ▷ $\mathcal{K}$ Container for the results
  for each T in $\mathcal{D}$ **do**
    $\Phi \leftarrow$ extract_noun_phrases(T)
    $\Phi \leftarrow$ filter(Φ)
    $\mathcal{S}_T \leftarrow$ categorize_s-phrases(Φ)
    $\mathcal{P}_T \leftarrow$ categorize_p-phrases(Φ)
    for each s in $\mathcal{S}_T$ **do**
      for each p in $\mathcal{P}_T$ **do**
        $v \leftarrow$ compute_idea_value(p, s)
        $idea \leftarrow \langle p, s, v \rangle$
        $\mathcal{K} = \mathcal{K} \cup \{idea\}$ ▷ No redundancy in $\mathcal{K}$
  return $\mathcal{K}$ ▷ The output known-idea set

To summarize the known idea construction algorithm: first extract noun-phrases, followed by noun-phrases classification (classify them into problem or solution) and then calculate the pair value.

Relevance values for *known-ideas*. In this study, four statistics (OCC, FREQ, CON, TF-IDF) were developed, which attempt to measure the relevance of a *known-ideas* to future recommendations (and which analogies to the *rating* a user has assigned to an item in traditional collaborative filtering). Each statistic is described below, with examples illustrated in Table 3.10:

TABLE 3.10: Examples of known-idea triplets from a sample corpus of 20 documents:

p-phrase	s-phrase	OCC	FREQ	CON	IF-IDF
global warming	climate model	3	24	0.21	19.77
neuropsychiatric disorders	mouse model	3	16	0.50	13.18
impurity atoms	three-dimensional atom probe technique	2	6	0.5	3.00
nickel catalysis	photoredox-metal catalysis approach	1	4	1.00	5.20
impulsive optical excitation	first-principle theoretical simulations	1	3	1.00	3.90

The values on the columns $OCC, FREQ, CON, IF-IDF$ indicate that the higher the value, the stronger relationship is between the problem and the solution.

OCC: the simplest way of estimating the relationship intensity between a *p-phrase* and *s-phrase* is to count the number of distinct documents in which they both appear (based on the assumption that the more times they co-occur, the stronger the relationship there is between them):

$$OCC(p,s) = \Big|\{T \in \mathcal{D} : p \in P_T, s \in \mathcal{S}_\mathcal{T}\}\Big| \tag{3.11}$$

The occurrence of a problem and solution pair only considers the number of documents that contain both the problem and solution phrases. Note that this experiment did not use semantic meaning comparison, that is to say the words (after stemming) which are exactly the same could be accumulated.

FREQ: idea frequency is similar to document occurrences, but also takes into account the frequency of idea-pair occurrences *within* documents:

$$FREQ(p,s) = \sum_{T \in \mathcal{D}} \Big|\{\langle p,s\rangle \in P_T \times S_T\}\Big| \tag{3.12}$$

CON: In order to address the fact that term counts alone cannot reflect the fact that some problems have numerous lines of attacks, while others have only a limited solution pool, the statistic *contribution* was defined . This is a normalization that divides the number of times a certain idea pair co-occurs by the total number of s-phrases used to address the same problem:

$$CON(p,s) = \frac{OCC(p,s)}{\Big|\{T \in \mathcal{D} : s \in S_T\}\Big|} \tag{3.13}$$

IF-IDF: *idea frequency / inverse document frequency* is an adaptation of the traditional *tf-idf* statistic that has been designed to addresses two observations: 1. the more times an *s-phrase* occurs in any given document, the more likely it is to be a 'key' solution to the document's *p-phrases*, so it is the one to be favoured; and 2. if an idea-pair crops up across the whole corpus the less likely it is to be "interesting" - either its research value has been saturated, or it is semantically redundant pairing in the same vein as a stop word. IF-IDF balances these two conflicting issues via the following formula[17]:

$$IF\text{-}IDF(p,s) = FREQ(p,s) \times \log\left(\frac{|\mathcal{D}|}{OCC(p,s)}\right) \tag{3.14}$$

Novel idea computation.

In this part, the prediction of new links between problems and solutions through comparison of *s-phrase* and *p-phrase* patterns that span across different domains was addressed. There are numerous ways to measure similarity between such patterns, but the strategy at the heart of the techniques is based upon a collaborative filtering (Breese et al., 1998). Collaborative filtering and the recommendation systems they underpin are based on imputation - a target user's past behaviours are first modelled and then compared to the habits of other users. Items that are favoured by similar users but do not yet exist in the target user's history, are then used as the basis for new recommendations. The technique considers *p-phrases* as analogs to users, and *s-phrases* as items. Traditional recommender systems can be formulated as *user-based* or *item-based* algorithms (Sarwar et al., 2001) and both approaches were assessed. In the generation of novel-ideas, the collaborative filtering task consists of the following steps:

1. **construct a preference matrix:** in this case, each row represents a *p-phrase* and each column represents an *s-phrase* with the numerical value at the intersection of a row and a column represents the idea's relevance value, v (as selected from one of the statistics in 3.2.1).

[17] N.b. Idea Frequency (IF) differs from traditional Term Frequency (TF) in that it counts the idea's support over the *whole* corpus, and not just for a single document, resulting in a global statistic.

2. **compute similarity scores**: for a specific problem (i.e. a row in the preference matrix), u, iterate through every other problem, w, and compute a similarity s between u and w and retain the k nearest neighbours, $\mathcal{N}$. In the experiments, N for each of the following distance metrics were optimized (refer to 2.2.2) : *Tanimoto*, *Loglikelihood* and *CityBlock*[18].

3. **generate novel-idea pairs:** this is achieved by recommending novel solutions to existing problems. For each potential solution, i, that the current problem has no entry for, every vector, w, was considered in the neighbourhood, N, and add its relevance score for solution i to a running average, weighted by the vector's similarity score s. Finally, results are sorted, producing a ranked list of novel *s-phrases* to the *p-phrase* under consideration. The top n *s-phrases* are combined with the p-phrase under consideration as the novel-idea prediction (in the experiments n is drawn from 2, 5, 10).

3.2.2 Idea detection using neural networks

Rule based method was adopted to classify the noun-phrases into problems and solutions in 3.2.1. Although <problem, solution> pairs embody an effective definition of ideas, the representation of problem and solution is not easily to be defined. While the primary concepts predominantly exist in the noun-phrases, simply using noun-phrases to represent problems and solutions is not enough. For example, from the sentence *researchers have developed a computational method to predict the function of unknown yeast genes*, simply using noun phrase *yeast genes* to represent research problem is not as clear as using a span of consecutive text *predict the function of unknown yeast genes*. In order to make the expression of the <problem, solution> more comprehensible, in this section, a supervised machine learning based method 2.1.2 was applied to idea mining: a new representation is proposed: a span of consecutive text to represent problem and solution. The method was evaluated on a human annotated dataset (refer to 3.1.2).

[18] please refer to http://mahout.apache.org for implementation details.

Detecting the Problem-Solution sequences (PSS) is effectively a form of Named Entity Recognition (NER) (E. F. T. K. Sang & Meulder, 2003) (refer to 2.1.4) rather than sentence classification, since only parts of the sentence were considered as the target to be tagged. In order to detect PSS, two models were explored. One of the models has been widely used for the task of NER. Inspired by the IOB (Cuong et al., 2014) format (short for Inside, Outside, Beginning), the goal is that for every word in the abstract, it should be correctly labelled as one of the five categories: outside of the chunk (O), beginning of the problem (B-P), inside of the problem (I-P), beginning of the solution (B-S) or inside of the solution (I-S). In the end, the words with the labels (B-P, I-P) are concatenated according to their orders in the original sentence to form the problem-sequence. Similarly, the solution-sequence is derived by concatenating the words with the labels (B-S, I-S). The five classes to be predicted are **B-P, I-P, B-S, I-S** and **O**, which stand for:

Problem-Beginning, Problem-Inside, Solution-Beginning, Solution-Inside and **Outside**.

These labels are defined by myself and the logic behind this definition is inspired by the task of NER.

An expected tagged sequences on the abstract (Hochreiter & Schmidhuber, 1997) is shown in the Figure 3.3: where the problem/solution sequences were highlighted in yellow/blue and the first word of the problem (solution) sequence would be classified as B-P(B-S)

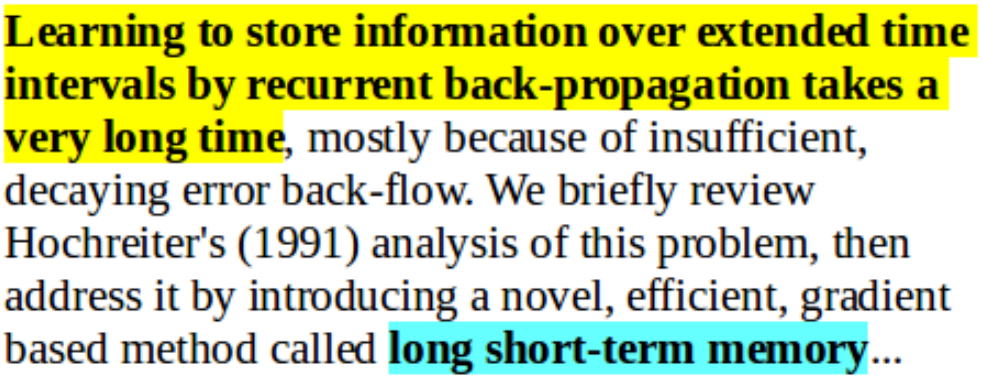


FIGURE 3.3: The *problem* (*solution*) sequences were highlighted in yellow (blue)

Models for Detecting the Scientific Ideas. Existing models for sequence labeling are linear statistical models, such as Maximum Entropy Markov models (MEMMs) (Mccallum et al., 2000) and Conditional Random Fields (CRF) (Lafferty et al., 2001). Research findings have shown that the model combining bidirectional LSTM (Bi-LSTM) networks and CRF is robust and it can produce accurate tagging performance without resorting to word embedding (Z. Huang et al., 2015; Lample et al., 2016). Considering that the task belongs to tagging problem and the words surrounding the problem and solution tags have certain patterns, it is hypothesized that using Bi-LSTM-CRF to detect problem and solution sequences can give better results. For comparison, a simple Neural Network with contextual word embeddings (CWE-NN) was used as a baseline.

Neural Networks with contextual word embeddings (CWE-NN). The model using a simple neural network was trained by utilizing the contextual embeddings of a word as the inputs to predict its label. The contextual word embeddings were constructed by first specifying a window, which frames the information surrounding the word, and then concatenating the word embeddings retrieved from the pre-trained word2vec models. Two models trained on Google News and scientific papers were compared. The scientific papers used for training the word2vec model are taken from ACL Anthology Reference Corpus [19], with any words that could not be found within the word2vec model not being considered. Resulting embeddings were used as the input features for updating the parameters in the neural network, which is expected to classify the words into the five categories accordingly.

The architecture of CWE-NN was implemented using Keras sequential model[20]. It is demonstrated in Figure 3.4.

For the first round experiment (150 train, 100 development and 100 test), the detailed parameters of the neural network are as follows: the activation function is ReLu, which outperformed other activation functions for several tasks (Krizhevsky et al., 2012). The input dimension is 1500, the output (the output from the input

[19] *http* : *//acl − arc.comp.nus.edu.sg*
[20] *https* : *//keras.io/models/sequential/*

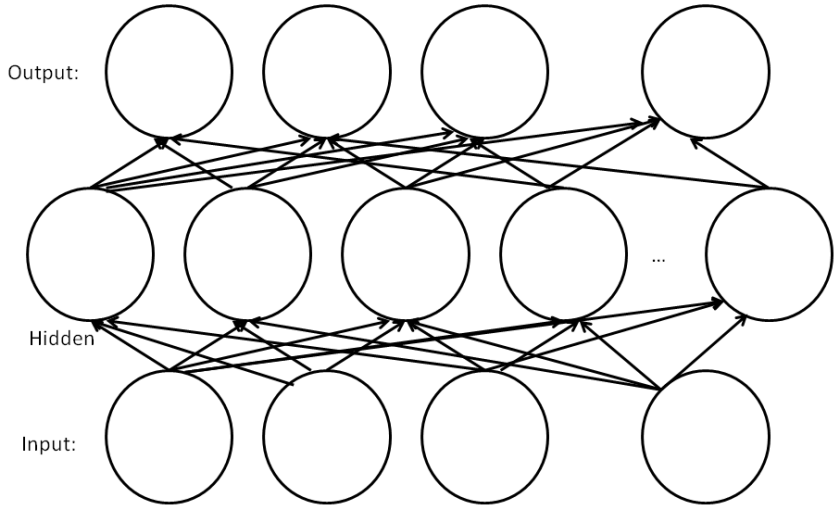


FIGURE 3.4: The architecture of the CWE-NN model, which is typical artificial neural networks.

goes to the hidden layer) dimension is 1024, the dropout rate is 0.5, the learning rate is 0.01 and stochastic gradient descent was used to update the weights. The loss function is categorical cross entropy because it is a multiclass classification problem. The optimizer used is the Adaptive subgradient method (Duchi et al., 2011). The input dimension is determined by the number of classes and the dimension of the word representation. The averaged precision of the five classes changed according to the different size of hidden layer nodes. Table 3.11 showed that when the size of the hidden layer nodes is set to 1024, the averaged precision of the five classes is 0.84, which is better than other settings.

hidden layer nodes	precision	recall
256	0.82	0.90
512	0.83	0.90
1024	0.84	0.90
2048	0.83	0.90

TABLE 3.11: Different settings of the hidden layer nodes resulted in different outcomes.

Another round of experiments focused on hyper-parameter optimization. A well-known hyper-parameter optimization method in deep learning is random-search

(Bergstra & Bengio, 2012). It's argued that for most data sets only a few of the hyper-parameters matter. Random-search independently draws from a uniform density from the same configuration space as would be spanned by a regular grid, as an alternative strategy for producing a trial set. In this experiment, random-search was conducted on CWE-NN model with wordvector trained on Google News. The best hyper-parameters combination is as follows: batch size is 105, dropout is 0.57, number of hidden layer: 797, optimizer is Adadelta (Zeiler, 2012), activation function is Relu (Nair & Hinton, 2010), number of epoch is 100. The best result (averaged precision is 0.85 and recall is 0.92) was obtained using this set of parameters.

Bidirectional LSTM and CRF tagging (Bi-LSTM-CRFs). Apart from CWE-NN, Bi-LSTM-CRFs was proposed since the advantages of Bi-LSTM-CRFs are (1) Bi-LSTM takes into account the information from both of the left and right side of the current word; (2) instead of predicting the label of the individual word independently, CRF has the transition matrix connecting the context with the current word. Let X be the word sequence to be tagged and Y be the labels. The bidirectional LSTM networks (bi-LSTMs) first convert the representation of a word in the sequence X into the format that is useful for the tagging task: $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$, which embeds the concatenation of the left and right context surrounding the word. Using the outputs from the bi-LSTMs, the conditional random fields (CRFs) learn the relationships between the observations (the input variables X and the outputs Y) and construct consistent interpretations. The formulations for training the model are listed bellow: (Lample et al., 2016):

For an input sentence:

$$X = (x_1, ..., x_n) \tag{3.15}$$

where x_i is the ith word representation, which is the vector. The word vectors can be derived from the embedding layers or from pre-trained word2vec model.

let P with the size nxk to be the matrix of scores output by the bidirectional LSTM network, where k is the number of distinct tags and $P_{i,j}$ is the score of the j^{th} tag of the ith word in a sentence.

For a sequence of predictions:

$$y = (y_1, ..., y_n) \tag{3.16}$$

its score is defined as follows:

$$s(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \tag{3.17}$$

where A is a matrix of transition scores such that $A_{i,j}$ represents the score of a transition from the tag i to tag j. y_0 and y_n are the *start* and *end* tags of a sentence. The matrix A is a square matrix with the size $k + 2$. The reason to add 2 is to take into account two extra labels (START and END). The transition matrix is initialized randomly with the size: (number of labels + 2) by (number of labels + 2). In this PSS tagging task, the number of labels is 5.

The transition matrix A is a parameter of a Bi-LSTM-CRF model. All the random assigned scores on A will be updated automatically during training process. That is to say, the CRF layer can learn the constraints by itself. There is no need to build the matrix manually. The scores will be more and more reasonable gradually with increasing training iterations [21].

A softmax over all possible tag sequences results in a probability for the sequence y:

$$p(y|X) = \frac{e^{s(X,y)}}{\sum_{\hat{y} \in Y_X} e^{s(X,\hat{y})}} \tag{3.18}$$

[21] $https://createmomo.github.io/2017/09/23/CRF_Layer_on_the_Top_of_BiLSTM_2/$

In the training process, the log-probability of the correct tags in the sequence was maximized :

$$log(p(y|X)) = s(X, y) - log\left(\sum \hat{y} \in Y_X e^{s(X,\hat{y})}\right) \tag{3.19}$$

The parameters to be optimized are in A and P, which are integrated in s.

With respect to a certain sentence X, Y_X represents all possible tag sequences. In the decoding process, the predicted output sequence was the one that has the maximum score:

$$y^* = argmaxs(X, y), \hat{y} \in Y_X \tag{3.20}$$

Input Representations and the mechanism of the system. At time t (when the algorithm is focusing on one word and trying to predict its label), the features are input into the system via the input layer. There are several options to represent the features, such as using one-hot-vector (Uriartearcia, 2014) or word embeddings (Mikolov, Chen, et al., 2013). The dimensionality of the feature size determines the size of the input layer. Word2vec (Mikolov, Chen, et al., 2013) has proved to be an effective representation in some of the NLP tasks, such as sentence classification (Komninos & Manandhar, 2016) and sentiment detection (Xue, Fu, & Zhan, 2014). In this experiment, the pretrained word embeddings that are fine-tuned were used. The dimension of the embeddings is 100, the minimum word frequency is 4 (if a word's occurrences is lower than 4 in the corpus, the word will not be considered in the experiment), and the window size is 8 (how the vector for one word is derived is affected by the window size, since how many words before and after a given word would be included as context words of the given word). The current word's left context and right context are represented by vectors, which are concatenated resulting in a representation of the current word. Figure 3.5 demonstrates the Bi-LSTM-CRFs framework that's adopted in this work. l_i (r_i) represents the word i and its left (right) context vector. The two vectors are concatenated forming the representation of the word i in the context c_i. The top layer is the CRF layer.

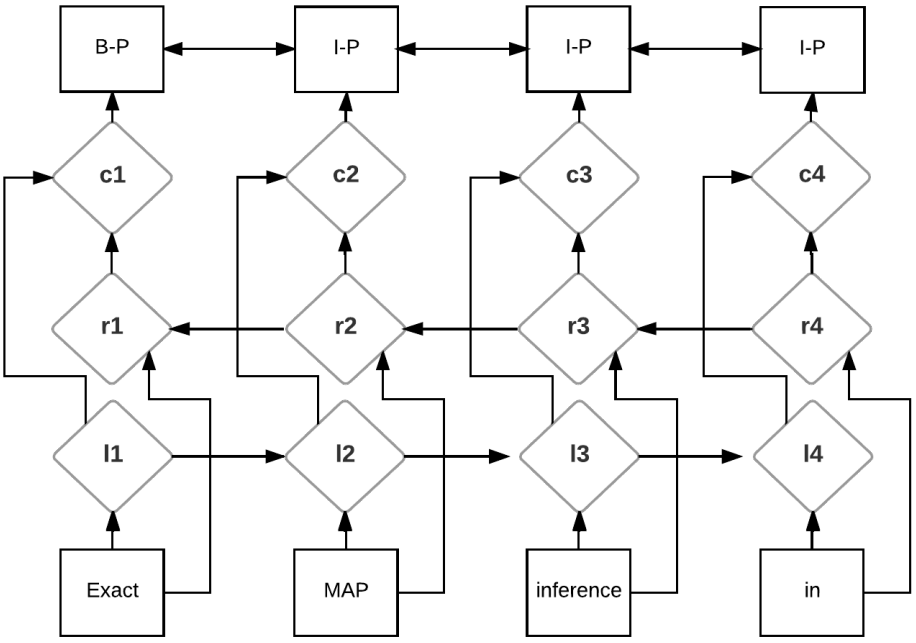


FIGURE 3.5: The architecture of the Bi-LSTM-CRF model. l_i (r_i) represents the word i and its left (right) context vector. The two vectors are concatenated forming the representation of the word i in the context c_i. The top layer (B-P, I-P, I-P, I-P) is the CRF layer. The arrows indicate the information (originally from the input text and carried out indirectly by matrix to the top layer) flow.

For Bi-LSTM-CRFs. The detailed implementation was based upon that described in (Lample et al., 2016). For updating the parameters in the networks, back-propagation was performed on the training data set. A stochastic gradient descent with a learning rate of 0.01 and a gradient clipping of 5.0 was used for optimizing the objective function.

For CWE-NN. The parameters used were as follows: the window size for building the contextual embeddings is 2; the word embedding dimension is 300; the learning rate is 0.1 and the dropout rate is 0.5.

3.2.3 Argumentative Zoning

Models. This experiment is aiming at learning sentence embeddings from large text corpus as features to classify sentences into seven categories in the task of AZ. Three models are investigated to obtain the sentence vectors: averaging the

vectors of the words in one sentence, paragraph vectors and specific word vectors. Following the word2vec introduction in 2.3.1. Before introducing word2vec, it is necessary to present *probabilistic language modeling*.

Probabilistic language modeling. The goal of probabilistic language modeling is to compute the probability of a sentence or sequence of words. An example of probability computation for the sentence $S = The\ man\ is\ feeding\ the\ dog.$ is demonstrated bellow:

Probabilistic language modeling :

$$P(S) = P(The, man, is, feeding, the, dog) \tag{3.21}$$

where P(The, man, is, feeding, the, dog) can be computed using chain rule (Ambrosio & Dal Maso, 1990):

$P(The, man, is, feeding, the, dog) =$
$P(The)P(man|The)P(is|The, man)P(feeding|The, man, is)$
$P(the|The, man, is, feeding)P(dog|The, man, is, feeding, the)$

The joint probability of words in the sentence can be expressed in the following formula:

Joint probability of words in sentence:

$$P(w_1 w_2 ... w_n) = \prod_i P(w_i|w_1 w_2 ... w_{i-1}) \tag{3.22}$$

It is important to find a solution to estimate those probabilities. Markov assumption (Paek & Chickering, 2005) was used for simplifying the computation of the joint probability:

Joint probability of words in sentence is simplified:

$$P(w_1 w_2 ... w_n) \approx \prod_i P(w_i|w_{i-k} w_2 ... w_{i-1}) \tag{3.23}$$

Ngram model (Kit et al., 2003) can help to further compute the joint probability. Bigram model has been widely used in NLP (Martin et al., 1998). A bigram is a sequence of two adjacent words from a sentence. The frequency distribution of the bigram in a piece of text (usually in a complete sentence) is commonly used for analyzing text. The conditional probability of a word given the preceding word can be computed using bigrams:

conditional probability:

$$P(w_n|wn-1) = \frac{P(w_{n-1}, Pw_n)}{P(w_{n-1})} \tag{3.24}$$

The conditional probability can be estimated by counting the occurrences of the bigrams:

conditional probability by counting:

$$P(w_n|wn-1) = \frac{count(w_{n-1}, Pw_n)}{count(w_{n-1})} \tag{3.25}$$

where the occurrences of each word can be obtained from text corpus. Different style of corpus give different patterns. For example, the occurrence of the bigram *was studied* is expected higher in scientific publications than in the news corpus. In this section, in order to train the word vectors, both news corpus and scientific papers were used. Although the scientific text corpus is more related to this task, the news corpus is expected to have its own advantage in that it is large and covers diversity of topics.

Based on the statistical language model introduced above, the mathematical explanations of word2vec (Le & Mikolov, 2014) are shown bellow:

Given a sequence of training words $w =< w_1, w_2, ..., w_n >$, the objective of the word2vec model is to maximize the average log probability of a sentence based on language model. in equation 3.26:

$$\frac{1}{T}\sum_{t=k}^{T-k} log(p(w_t|w_{t-k},...,w_{t+k})) \tag{3.26}$$

Using softmax technique, the prediction can be formalized as:

$$p(w_t|w_{t-k},...,w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{3.27}$$

That is to say that using maximum Likelihood principle to maximize the probability of the word w_t (for *target*) given the surrounding words (left and right of the word w_t). The range of i is the size of the vocabulary in the training corpus.

Each of y_i is un-normalized log probability for each output word i:

$$y = b + Uh(w_{t-k},...,w_{t+k};W) \tag{3.28}$$

where U, b are the softmax parameters. h is constructed by a concatenation or average of word vectors extracted from W. The vectors to represent words are obtained once the training converges. In the end, every word is mapped to a unique vector, represented by a column in a matrix W. The detailed explanation of U, b can be found in the paper (Tang et al., 2015).

As stated in the paper (Kenter et al., 2016), simply averaging word embeddings of all words in a piece of text has proven to be a strong baseline for many tasks (Faruqui et al., 2015; L. Yu et al., 2014), that is why the first model being explored is averaging word vectors ($AVGWVEC$): to average the vectors in word sequence $w =< w_1, w_2, ...w_n >$. The main process in this model is to learn the word embedding matrix W_w:

$$V_{avgwvec}(w) = \frac{1}{n}\sum W_w^{x_i} \tag{3.29}$$

where W_w is the word embedding for word x_i, which is learned by the classical word2vec algorithm (Mikolov, Chen, et al., 2013).

The second model, $PARAVEC$, is aimed at training paragraph vectors. It is also called distributed memory model of paragraph vectors (PV-DM) (Le & Mikolov, 2014), which is an extension of word2vec. The concept of distributed representations for words is not new. It was first introduced by Rumelhart et al (Rumelhart et al., 1986) in 1986 and it has proven to be successful for statistical language modeling. The methods introduced by Socher et al (Socher, Lin, et al., 2011; Socher, Pennington, et al., 2011) have disadvantage that it requires parsing and it is based on supervised learning, which needs a lot of labeled data for training.

The third model is constructed for the purpose of improving classification results for a certain category. In this study specifically, improving the results of identifying BAS sentences in AZ is the main focus. In this experiment, BAS specific word embeddings ($BSWE$) were trained, which was inspired by Tang et al. (Tang et al., 2014)'s model: Sentiment-Specific Word Embedding (unified model: $SSWE_u$). They developed special neural networks for incorporating the supervision from sentiment polarity of text, such as sentences or tweets, in their loss functions, thus, sentiment information in the continuous representation of words are encoded. In order to obtain large scale training dataset, they have trained the sentiment-specific word embedding from massive distant-supervised tweets, which were collected by scanning the positive and negative Emojis. For example, if there is a smiling face emoticon in a tweet, a positive sample would be obtained and this tweet will be sent to the neural network for training (as positive samples). In this experiment, similar approach is explored. A collection of sentences that have been retrieved by cue words matching (there are hand selected positive and negative cue words) are sent to the neural network for training the sentiment specific word vectors. After obtaining the word vectors via $BSWE$, the same scheme to average the vectors in one sentence as in the model $AVGWVEC$ was used.

Classification and evaluation. The learned word embeddings are input into a classifier as features under a supervised machine learning framework, Support

Vector Machine (SVM). Similar to sentiment classification using word embeddings (Tang et al., 2014), where they try to predict each tweet to be either positive or negative, in the task of AZ, the embeddings to classify each sentence into one of the seven categories were used.

To evaluate the classification performance, precision, recall and F-measure were computed. To compare the three models effectiveness on the AZ task, each model was trained on a same ACL dataset (introduced int he dataset section). The word2vec was trained using different parameters, such as different dimension of features. The first model was trained on different corpus to evaluate the impact from different domains.

Parameters. Inspired by the work from Sadeghian and Sharafat (Sadeghian & Sharafat, n.d.) [22], the word to vector features were set up as follows: the Minimum word count is 40; The number of threads to run in parallel is 4 and the context window is 10. The explanation of context window is stated in 2.3.1.

Strategy of dealing with imbalanced data. In imbalanced data sets, some classes are significantly outnumbered by other classes (H. M. Nguyen et al., 2011), which affects the classification results. In this experiment, the test dataset is an imbalanced data set. Table. 3.12 shows the distribution of rhetorical categories from the *AZ* test dataset. The categories OWN and OTH are significantly outnumbering other categories.

TABLE 3.12: Distribution of rhetorical categories

Category	Number of Sentences	Percentage
OWN	4868	0.54
OTH	1927	0.21
BKG	644	0.07
BAS	641	0.07
CTR	451	0.05
AIM	303	0.03
TXT	191	0.02

To deal with the problem of classification on imbalanced data, Synthetic Minority Over-sampling TEchnique (SMOTE) (Chawla et al., 2002) were performed on the

[22] *https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-2-word-vectors*

original dataset [23]. 10-fold-cross validation scheme was adopted and the results were averaged from 10 iteration.

3.2.4 Citation sentiment analysis

Pre-processing. I used SentenceModel provided by LingPipe to segment raw text into its constituent sentences[24]. The data that has been used to train the vectors has noise. For example, there are incomplete sentences mistakenly detected (e.g. Publication Year.). To address this issue, sentences with less than three words were eliminated.

Overall Sent2vec Training. In this work, sentence embeddings based on word embeddings were explored. Simply averaging the vectors of the words in one sentence to obtain sentence embeddings (sent2vec) was evaluated. The parameters that has been used to train the word embeddings are the same as in the work of Sadeghian and Sharafat (Sadeghian & Sharafat, n.d.): the word to vector features were set up as follows: the Minimum word count is 40; The number of threads to run in parallel is 4 and the context window is 10. The explanation of context window is stated in 2.3.1.

Polarity-Specific Word Representation Training. To improve sentiment citation classification results, polarity specific word embeddings (PS-Embeddings) were trained, which were inspired by the Sentiment-Specific Word Embedding (Tang et al., 2014). After obtaining the PS-Embeddings, the same scheme to average the vectors in one sentence according to the sent2vec model was used.

Directly inspired by (Tang et al., 2014), the idea behind PS-Embeddings is described bellow :

Given an original ngram and the sentiment polarity of a sentence as the input, PS-Embeddings predicts a two-dimensional vector for each input ngram. There

[23] The synthesized samples need to be further examined in terms of if they are meaningful and representative.

[24] `http://alias-i.com/lingpipe/docs/api/com/aliasi/sentences/SentenceModel.html`

are two scalars, one of which is representing language model score and the other one stands for sentiment score of the input ngram. The training objectives of PS-Embeddings are that on one hand the original ngram should obtain a higher language model score and on the other hand the sentiment score of original ngram should be more consistent with the gold polarity annotation of sentence than corrupted ngram. The loss function of PS-Embeddings is the linear combination of two hinge losses, which are detailed in the Equations (8) and (9) in the paper (Tang et al., 2014).

The polarity sentences are originally from manually extracted 82 polar phrases from citation sentences in the ACL Anthology Network corpus (Athar, 2011). [25]

Classifier.

Support vector machines (SVMs) are used to train the classification model.

3.2.5 Idea flow analysis

Using natural language processing and machine learning techniques, researchers investigated the methodologies to classify the citation sentences into three categories: objective, positive and negative (Athar, 2011). Collections of annotated corpus for the task of citation sentiment analysis (CSA) were created[26]. Despite the effectiveness on the task of CSA, the above works can only capture the polarity of a single citation sentence towards the cited paper. To evaluate a citation's impact on an idea, analyzing a single citation is not enough. Taking all the citations that cite the same paper in one publication may give an overall sentiment score of the cited paper, but evaluating the sentiment changes of the citing papers that cited the same paper over time is also important. Because the changes may reflect the impact of the cited paper from fluctuation perspective. By fluctuation, it refers to the sentiment changes between positive and negative. It's assumed that ideas with different fluctuation levels have different properties. For example,

[25] $http://cl.awaisathar.com/citation-sentiment-corpus/polar_phrases.txt$
[26] http://cl.awaisathar.com/citation-context-corpus/

if the ideas are with strong fluctuation, it may indicate that the problem discussed in that paper is a hot topic, or the solution proposed in that paper needs to be improved. Thus, study idea flow may help researchers to find research gaps. This experiment is aiming at answering the questions:

- If the author(s) referenced a paper (P_C) multiple times in a specific paper (P_A), what is the overall sentiment of the paper P_A towards P_C?
- If the paper P_C has been cited by different papers P_A and P_B, assuming P_B was published earlier than P_A, how to model the citation sentiment changes (with respect to the paper P_C) over time[27]?

These issues are tackled using statistical analysis - analyzing the sentiment level co-occurrence matrix. As follows:

- Both the Internal Citation Sentiment Sequence (ICSS) and the Global Citation Sentiment Sequence (GCSS) were analyzed manually and automatically. ICSS focuses on citations originating from within a single paper, whereas GCSS is more concerned with multiple papers.
- Two scores were computed for analyzing the ICSS: the difference between the positive and negative citation counts; the ratio of the positive citations to the total citations.
- In addition to the two scores, sentiment Level Co-occurrence Matrix (SLCM) was proposed for classifying the ICSS.
- Eddy Dissipation Rate (EDR) was applied for calculating the GCSS turbulence level. The details about EDR refer to 2.3.6

Observations. The visualized sentiment citations used for this work is a corpus consisting of manually labeled citation sentiment[28]. These enable readers to observe the citation sentiment changes by providing different colors that represents

[27]The length of time of the fluctuation is important. This is why when comparing global citation sentiment changes, only citation sequences covering relatively same year-span are considered. For example, in Figure 3.9

[28]$http://cl.awaisathar.com/citation-context-corpus/$

the sentiment of the citation sentence. The data presented in the HTML files contains all papers in the ACL Anthology Network (AAN) which cite a specific target paper. On their website, each row represents a citing paper and each square in that row represents a sentence in the citing paper. The colour of each square indicates the sentiment of the sentence. When hovering on the square, a citation sentence will pop up.

Examples of the citing and cited papers in html format are demonstrated in the following box. In this experiment, the citation sentiment changes were analyzed statistically. The citations with the ground truth sentiment are necessary to conduct the experiment, thus, it is relying on the expert annotated corpus (Athar & Teufel, 2012a) that have the citations ordered by the publication time.

<td class="line x" title="1:153 Determining the Sentiment of Opinions Soo-Min Kim Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, CA 90292-6695 skim@isi.edu Eduard Hovy Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, CA 90292-6695 hovy@isi.edu Abstract Identifying sentiments (the affective parts of opinions) is a challenging problem." ></td>
<td class="line x" title="2:153 We present a system that, given a topic, automatically finds the people who hold opinions about that topic and the sentiment of each opinion." ></td>
<td class="line x" title="3:153 The system contains a module for determining word sentiment and another for combining sentiments within a sentence." ></td>
<td class="line x" title="4:153 We experiment with various models of classifying and combining sentiment at word and sentence levels, with promising results." ></td>
<td class="line x" title="5:153 1 Introduction What is an opinion?" ></td>
<td class="line x" title="6:153 The many opinions on opinions are reflected in a considerable literature (Aristotle 1954; Perelman 1970; Toulmin et al. 1979; Wallace 1975; Toulmin 2003)." ></td>
<td class="line oc" title="7:153 Recent computational work either focuses on sentence subjectivity (Wiebe et al. 2002; Riloff et al. 2003), concentrates just on explicit statements of evaluation, such as of films (Turney 2002; Pang et al. 2002), or focuses on just one aspect of opinion, e.g., (Hatzivassiloglou and McKeown 1997) on adjectives." ></td>

Citation Sentiment Changes. Inspired by Nguyen et al. (L. T. Nguyen et al., 2012), the citation sentiment changes between positive and negative are with higher priority, thus, the citations with neutral polarity are ignored. Citation sentiment change is defined in such a way that it counts for one citation sentiment change if the citation sentiment polarity is changed from positive to negative or from negative to positive. Examples reflecting the citation sentiment changes are demonstrated in the Table 3.13. Even though the paper *W06-1615* was being

TABLE 3.13: Observations of citation sentiment changes - Internal

CitedP	CitingP	Citations	Polarity
W06-1615	E09-3005	SCL has been applied successfully in NLP for Part-of-Speech tagging and Sentiment Analysis.	1
W06-1615	E09-3005	An attempt was made in the CoNLL 2007 shared task to apply SCL to non-projective dependency parsing (Shimizu and Nakagawa, 2007), however, without any clear conclusions.	-1
W06-1615	E09-3005	We report on our exploration of applying SCL to adapt a syntactic disambiguation model and show promising initial results.	1

cited by the same author in the same paper *E09-3005*, the sentiment is different when citing completely different parts of the paper. For example, the first record on the Table 3.13 showed one piece of the ideas of the paper *W06-1615* was mentioned as the positive reference. The idea Structural Correspondence Learning (SCL) (Plank, 2009) was used as a solution for solving Part-of-Speech tagging and sentiment analysis problems by previous researchers. The second record expressed the unclear conclusion after applying SCL to non-projective dependency parsing problem[29]. In third record on the table, the same method SCL was referenced as a positive solution to adapt a syntactic disambiguation model. Because the results are promising, this citation was labeled as positive. These examples indicate that the citing paper tend to investigate several aspects of the same idea. Whether the idea is useful or not depends on the detail situation. It is reasonable to give a sentiment degree to the citation, which is similar to fuzzy set theory.

From the citation sentences shown in Table 3.13, it is observed that the research problem (discovered in the cited paper) and the problem solution (proposed in the

[29]The author of the paper (Athar & Teufel, 2012a) labeled this citation as negative, maybe because there is *however* in the sentence.

TABLE 3.14: Examples of citation sentiment sequences generated from one paper.

CitedP	CitingP	Citation Sentiment flow	Sentiment
P07-1033	D08-1105	1 1 1 1 1 1 1 1	Positive
A92-1018	W98-1110	-1 -1 -1 -1 -1 -1	Negative
W02-1011	P09-1028	1 -1 1 1 -1	controversial
J90-1003	D08-1007	1 -1 -1 -1 1 -1 -1 1 -1	controversial

citing paper) were described .

Citation Sentiment Sequence.

- Internal Citation Sentiment Sequence (ICSS).

  Let $X = \langle x_1, x_2, \ldots, x_i, \ldots, x_N \rangle$, be a sequence of sentences (X) that cited the paper P_C in the citing paper P_A, represented as an ordered sequence of N citation sentences[30], then the ICSS S_l is defined as follows: $S_l = (sl_1, sl_2, \ldots, sl_i, \ldots, sl_N)$, where $sl_i \in \{0, 1, -1\}$ expressing the sentiment polarity $\{objective, positive, negative\}$ in the citation x_i. An example of ICSS is *1 -1 1 1 -1*. More examples of ICSS with its computed sentiment (sl_i) are shown in the Table 3.14.

  The citation sentiment was annotated manually by Athar (Athar & Teufel, 2012a). They used a subset of the dataset from Athar (Athar, 2011), which consists of 20 target papers. These 20 papers correspond to approximately 20% of incoming citations in the original dataset. They contain a total of 1,555 citations from 854 citing papers. They used a four-class scheme for annotation. Every sentence which does not contain any direct or indirect mention of the citation is labelled as being excluded from the context. The rest of the sentences are marked either positive (1), negative (-1) or neutral (0).

- Internal Citation Sentiment Overall Score (ICSOS).

[30]Ordered by the citation's position in the paper. The citation appeared earlier in the paper holds the lower index in the set X

Based on the ICSS S_l, an overall sentiment score was derived that maybe possible to reflect the general sentiment polarity in the citing paper P_A towards the cited paper P_C. Two schemes to compute the score were investigated: simply adding up the sl_i, named $countcitations_l$ (equation 3.30), from which, a set of scores $Score_{rc} \subseteq \mathbb{Z}$ were obtained; positive citation ratio, named $pratio_l$, which will result in a set of scores $Score_{pr} \subseteq [0, 1]$. The methods for computing the scores are detailed in the section 3.2.5.

By analyzing the ICSOS, it is hypothesised that the following categories can help to classify the overall sentiment contained in the cited paper with regards to a specific citing paper: uncontroversial positive (P), uncontroversial negative (N), controversial (con). The sentiment categories shown on the tables were annotated by researchers manually.

- Global Citation Sentiment Sequence (GCSS).

  Let D be the document set that have cited the same paper P_C.

  $D = \langle d_1, d_2, \ldots, d_i, \ldots, d_N \rangle$,

  Note that they are ordered by published time.

  then the GCSS S_g is defined as :

  $S_g = (sg_1, sg_2, \ldots, sg_i, \ldots, sg_N)$,

  where $sg_i \in \mathbb{Z}$ expressing the overall sentiment degree in the citing paper d_i. Similar to computing the $countcitations_l$ (equation 3.30) and $pratio_l$ (equation 3.31), the $countcitations_g$ (equation 3.32) and $pratio_g$ (equation 3.34) are the two forms of sg_i. $countpapers_g$ (equation 3.33) and $pratiopaper_g$ (equation 3.35) were computed to measure the GCSS sentiment. The formulations are described in the section 3.2.5.

  The connection between sg and sl is that sg is the results of integrating all the sl regarding one cited paper in the same citing paper. There are several ways to integrate all the opinions toward the cited paper in the citing paper, namely, counting the negative and positive citations and do the calculation in equation 3.30 and calculating the ratio of the positive ones among all the citations, refer to equation 3.31.

TABLE 3.15: Examples of citation sentiment sequences (shown in vertical) generated from a collection of papers that cited the same paper. $G-1$ to $G-2$ in equation 3.31 and 3.32 represent the methods for computing the overall sentiment score described in section 3.2.5

CitedP	CitingPs (year)	$G1$	$G2$	Category
P04-1041.html	P04-1047 (2004)	1	1	P
	N06-1019 (2006)	1	1	
	W08-1306 (2008)	1	0.5	
N03-1003.html	W03-1602 (2003)	-1	0	N
	C04-1051 (2004)	-1	0	
	N04-1031 (2004)	-3	0	
	W04-0910 (2004)	-1	0.33	
	I05-5001 (2005)	-1	0	
P07-1033.html	C08-1003 (2008)	1	1	controversial
	D08-1105 (2008)	8	1	
	P08-1029 (2008)	-1	0	
	E09-1006 (2009)	2	1	
	E09-3005 (2009)	2	1	
W06-1615.html	D07-1129 (2007)	-2	0	controversial
	P07-1056 (2007)	-5	0.14	
	W07-2202 (2007)	-1	0	
	D09-1158 (2009)	2	0.75	
	E09-3005 (2009)	2	0.58	

Similarly, the connection between $countcitations_g$ and $countcitations_l$ is that the results of integrating all the $countcitations_l$ regarding one cited paper over the years [31].

The connection between $countcitations_l$ and sg is that one form of calculations of sg is using $countcitations_l$. The other form doesn't use $countcitations_l$ directly, it is using $pratio_l$ instead.

Note that $countcitations_l$, $countcitations_g$, $pratio_l$ and $pratio_g$ are invariant to the order of their components in the sequence. In order to capture the effects of the changes in the calculated list of scalars, other techniques are in used, such as SLCM introduced in 2.3.5.

- Global Citation Sentiment Overall Score (GCSOS). To measure the sentiment changes toward a cited paper within a specific period of time[32], GCSOS

[31] the limitation is, the corpus that has been obtained may not cover all the papers that have cited the examined paper.

[32] The time span from the first until the last time the paper was cited was used.

is needed. Four metrics are employed - (1) add up all the $countcitations_l$ from each of the citing paper (see equation 3.32); (2) compute the number difference between the positive citations and the negative citations: $countpapers_g$ (equation 3.33); (3) the positive ratio based on equation 3.31, that is $pratio_g$ in equation 3.34; (4) the positive ratio based on the overall number of positive and negative citing papers, that is $pratiopaper_g$, which is shown in equation 3.35.

Overall, the metrics are described in the section 3.2.5, which are detailed in equation 3.30 - 3.35.

Citation Sentiment Sequence Analysis Over Time. The purpose of analyzing the citations sentiment sequence over time is to uncover some useful statistics that cannot be obtained from the single observation. The paper (Y. Yin & Wang, 2017) studied how time affects citations and combined retrospective and prospective approaches to measure the citation age distributions. For global sentiment flow analysis, the task of Measuring the fluctuation in the GCSS was explored:

I argue that there is difference between:

S_1 = [-1, -1, -1, -1, -1, 1, 1, 1, 1, 1] and

S_2 = [-1, 1, -1, 1, -1, 1, -1, 1, -1, 1], even though the $countcitations_l$ of these two sequences are the same (0). It is possible that when the sentiment jumps between positive and negative often (in the GCSS) indicates the cited paper got controversial judgments from time to time. Inspired by the methods for estimating turbulence intensity from water droplets (Nijhuis et al., 2016), seven metrics were adopted as the citation sentiment turbulence indicator. The seven metrics are non-periodic variance method EDR, periodic 3rdorder EDR, non-periodic 3rdorder EDR, periodic power spectrum EDR, non-periodic 2ndorder EDR, periodic 2ndorder EDR, non-periodic power spectrum EDR. The implementation of the metrics can be found in the Python Module provided by Albert Oude Nijhuis [33]. The seven metrics were treated as seven features for the cluster algorithm.

[33] *https://github.com/albertoudenijhuis/edrlib*

Citation Sequence Score Computation.

- Measurement IN-1: $countcitations_l$ For a specific cited paper, let N_{pos} and N_{neg} be the number of positive and negative citations in a citing paper:

$$countcitations_l = N_{pos} - N_{neg} \tag{3.30}$$

- Measurement IN-2: $pratio_l$

$$pratio_l = N_{pos}/(N_{pos} + N_{neg}) \tag{3.31}$$

  Note that the + and − operations are addition and subtraction in scalar model.

- Measurement G-1: $countcitations_g$

$$countcitations_g = \sum_{i=1}^{n} countcitations_{li} \tag{3.32}$$

  where n represents the total number of papers that cited the specific paper. The purpose of divided by n is performing normalization, because the total number of citations (with respect to cited paper) varies.

- Measurement G-2: $countpapers_g$

  For a specific cited paper, let M_{pos} and M_{neg} be the overall number of positive and negative citing papers:

$$countpapers_g = M_{pos} - M_{neg} \tag{3.33}$$

- Measurement G-3: $pratio_g$

$$pratio_g = \sum_{i=1}^{n} pratio_l/n_i \tag{3.34}$$

  where n represents the total number of papers that cited the specific paper.

- Measurement G-4: $pratiopaper_g$

$$pratiopaper_g = M_{pos}/(M_{pos} + M_{neg}) \tag{3.35}$$

Figure 3.6 presents the process to count the occurrences for the sequence [1, -1, 1, 1, -1] using white-black prototype. This is a sliding comparison where the feature prototype is first compared with the first elements of the citation flow and then in subsequent steps it is compared with other element in the sequence. This figure shows how the occurrences of white-black pattern is increased from 1 to 2 through after four steps of sliding. Based on the occurrences, sentiment level co-occurrence matrix is created, which is shown in Figure 3.7. Four cells on the matrix indicate the number of the four feature prototypes. The top left cell with number 7 demonstrate the occurrence for the prototype white-white (1,1). The top right cell with number 0 demonstrate the occurrence for the prototype white-black (1, -1). The bottom left cell with number 0 demonstrate the occurrence for the prototype black-white (-1, 1). The bottom right cell with number 0 demonstrate the occurrence for the prototype black-black (-1, -1).

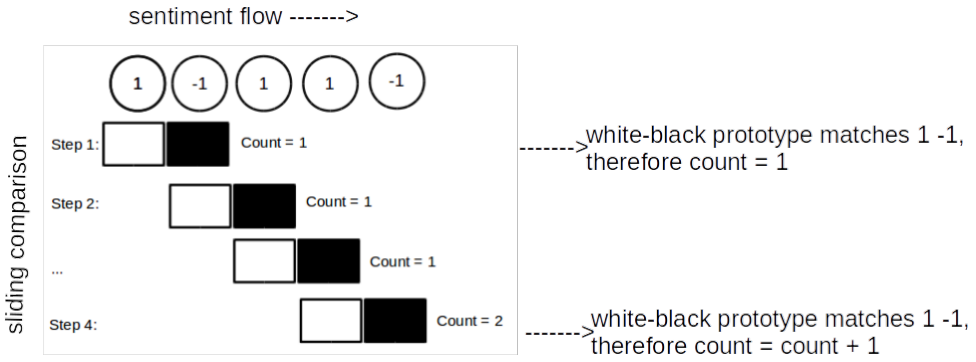


FIGURE 3.6: Using white-black prototype to analyze sequence [1, -1, 1, 1, -1].

The normalized SLCM for the sequence [1, -1, 1, 1, -1] is shown in Figure 3.8

The normalized vectors are obtained from the matrices shown in Figure 3.7 in such a way that the values in the matrix are concatenated from the top left cell to the bottom left cell in a clockwise fashion. The reason for concatenating the cells in such a way is that the SLCM can be seen as stochastic Matrix. A stochastic matrix, also called a probability matrix, probability transition matrix, transition

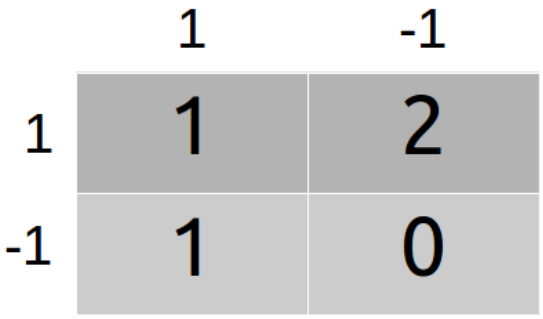


FIGURE 3.7: Sentiment level co-occurrence matrix for the sequence [1, -1, 1, 1, -1]. The pair of cells in a row reflect the sentiment change: the [1, 1] and [-1, -1] pattern stands for there is no sentiment change and both of the adjacent citations are positive or negative; the [1, -1] pattern represents the sentiment change from positive to negative and the [-1, 1] pattern means the sentiment changes from negative to positive.

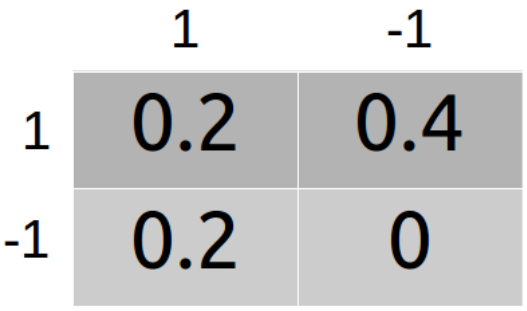


FIGURE 3.8: Normalized sentiment level co-occurrence matrix for the sequence [1, -1, 1, 1, -1].

matrix, substitution matrix, or Markov matrix, is matrix used to characterize transitions for a finite Markov chain, elements of the matrix must be real numbers in the closed interval [0, 1] [34]. Since in this experiment, the SLCM has the size of 2 by 2, it is computationally affordable to do matrix classification by flattening the matrix into one single vector, rather than using other matrix comparison methods (Cohen et al., 1998).

The vector (0.2, 0.4, 0, 0.2) is the normalized one of the original vector in Figure 3.7. The normalized vector was named as $SLCM_V$.

Eddy Dissipation Rate (EDR) for Measuring Fluctuation in GCSS.

To analyze the citation sentiment sequence over time, four GCSSs was plotted using the score $pratio_l$. The horizontal-axis represents the citations (over the

[34] $http://mathworld.wolfram.com/StochasticMatrix.html$

years) with respect to a specific cited paper and the vertical-axis represents the normalized sentiment level ($pratio_l$). Four GCSSs are shown in the Figure 3.9. According the observation on the plots, the top two GCSSs in Figure 3.9 are assumed to be less controversial due to the reason that given the same length of the sequence, the top two bar-charts in Figure 3.9 have more bars with consecutive zeros. In contrast, there are more non-zero bars in the bottom two charts in Figure 3.9. It is possible that controversial papers hold the potential for generating novel ideas since researchers can discover more research gaps in the controversial papers. On the contrary, uncontroversial papers are either used as a solution for solving new problems or discarded when new approaches emerged. According to different preferences from different people, they may selectively choose the papers to read.

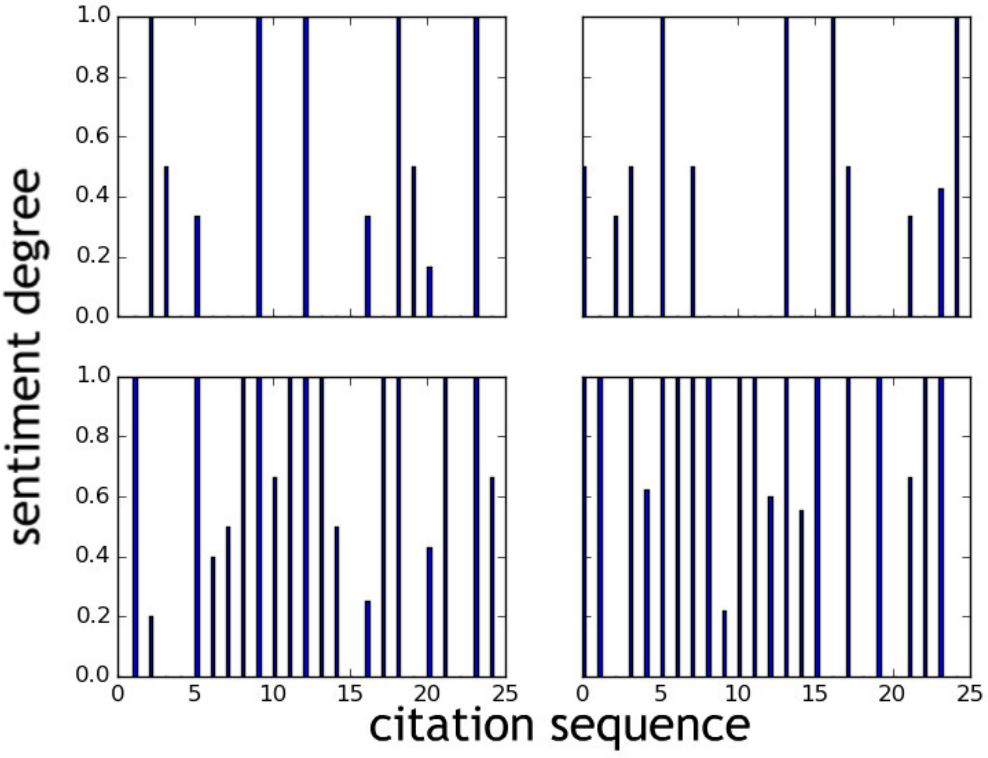


FIGURE 3.9: Plots of GCSSs (The cited-paper IDs are : J93-1007 (top-left), J90-1003 (top-right), W05-0909 (bottom-left) and W04-1013 (bottom-right). where the horizontal-axis represents the length of the sequence and vertical-axis is the normalized sentiment value $pratio_l$ (unnormalized ones are computed by Equation 3.30). The range of the value is 0-1, where 1 indicates strong positive and 0 indicates strong negative. The place without bar means the The $pratio_l$ value is 0. Clearly, the bottom two sequence are subjectively controversial. Four sequences with detailed $pratio_l$ values are shown in the Table 3.7.

According to the observations on the citation sentiment changes, the flows of the citation sentiment changes are turbulent. There are GCSSs, which are relatively less turbulent in comparison with others. To classify the papers into controversial and uncontroversial, the measurement called Eddy Dissipation Rate (EDR) (Cohn, 1995) was explored to statistically analyze the fluctuation level in the GCSS (the EDRs based on $countcitations_l$ and $pratio_l$ were compared). The reason to choose EDR as the method to analyze GCSS is that EDR has been proved to be the commonly-used turbulence intensity metric (J. Williams et al., 2006).

The sequence of citation sentiment degrees (either represented by $countcitations_l$ or $pratio_l$) is similar to the sequence of the air velocities, which are used for computing the EDR. Since there does not exist a unifying theory of turbulence, the approach used in this experiment is directly inherited from the method introduced in the paper (Nijhuis et al., 2016), which is originated from the work (Siebert et al., 2006). According to turbulence theory, a turbulence flow can be interpreted as a population of several eddies with different sizes and strengths. Two variables are important: the characteristic diameter of the eddies and their characteristic orbital velocity. Due to the limited dataset (20 GCSSs), it is inappropriate to use supervised learning approach to learn a sentiment fluctuation model. By fluctuation model, I refer to the model that can determine the fluctuation degree. Instead, cluster method was used[35] to discover the different fluctuation patterns in the GCSSs. The cluster number was set to two, representing controversial (1) and uncontroversial (0). [36] The features for clustering the data were represented by the seven metrics.

Citation Sequence Sentiment Classification.

Supervised learning for the sentiment classification. A Support Vector Machine (SVM) (Joachims, 1998), with 4-fold cross validation, was trained as a model to analyse this data. SVMs are effective supervised machine learning models.

[35] $http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html$

[36] It's not defined by description. It's more towards making decisions by the latent features, such as the EDR features.

The input features are composed of $SLCM_V$, $countcitations_l$ and $pratio_l$. To evaluate the individual feature's performance, the same scheme to train the separate models using each feature separately was used.

Chapter 4

Results and Discussion

This chapter contains the result and discussion with respect to each method that have been introduced in 3.2.

4.1 Idea mining and novel idea computation

4.1.1 Result

Using novel idea computation algorithm (refer to 3.2.1), 90,212 unique <*p-phrase*,*s-phrase*> known-idea pairs were constructed. The restricted number of known-ideas is because no cogent s-phrases could be extracted for many abstracts, even though numerous noun-phrases were identified. Examples of the system's output in Table 4.1 illustrate the cogent recommendations the system can produce. Precision and recall results are similarly positive - full results in Tables 4.2-4.4 indicate how the number of items in the recommendation set influences results for each of the relevance values that have been tested (with the fixed size of the neighbourhood (50) for each recommendation set size).

For each model parametrization a set of novel idea candidates were generated from the training set (46,250 pairs from 1998-2009). These were then evaluated to determine if recommended idea-candidates *actually occurred* (the problem and

solution phrases appear in the same abstract) in test set (43,962 pairs from 2010-2015), with results being summarized for each *p-phrase* using traditional precision, recall and F-measure scores. This process was repeated for both user- and item-based collaborative filtering, using relevance value statistics drawn from {OCC, CON, IF-IDF} (n.b. FREQ is not reported due to its similarity to OCC) and varying the size of the recommended solution list for each p-phrase ($n \in \{2, 5, 10\}$).

TABLE 4.1: Examples of novel idea-pairs generated from journal Science.

problem	old-solution	proposed-solution
large void volumes coincides	diffraction analysis	thermodynamic analysis
endoplasmic reticulum protein	systematic analysis	comparative genomic strategy
ancestral state reconstruction	likelihood-based approach	fluorescence technique
metallic single-wall carbon nanotube molecules	luttinger-liquid mechanism	surface chemical analysis
primary dendritic cells	unbiased approach	genome-wide location analysis

While the statistical results shown in Tables 4.2-4.4 are better than randomly generated ones. Each problem in training set was randomly recommended with 5 solutions and repeat this process for 5000 times. The averaged precision, recall, F measure are 0.000761, 0.00141, 0.000831.

TABLE 4.2: OCC performance (precision/recall/F-measure)

metric	2-recommendations	5-recommendations	10-recommendations
user Loglikelihood	0.0615/0.0407/0.0415	0.051/0.0479/0.0389	0.0486/0.0502/0.0371
user CityBlock	0.0187/0.015/0.0147	0.0179/0.0177/0.0149	0.0179/0.0177/0.0149
user Tanimoto	0.0635/0.0381/0.0404	0.0582/0.0475/**0.0416**	0.0572/0.0504/0.0412
item Loglikelihood	0.0141/0.00895/0.00957	0.0116/0.0167/0.0118	0.0104/0.0278/0.0128
item CityBlock	0.0122/0.00593/0.00648	0.00899/0.0126/0.00852	0.0079/0.022/0.00975
item Tanimoto	0.0176/0.00978/0.0109	0.013/0.0186/0.0131	0.0113/0.0307/0.0139

TABLE 4.3: CON performance (precision/recall/F-measure)

metric	2-recommendations	5-recommendations	10-recommendations
user Loglikelihood	0.061/0.0403/0.0406	0.0511/0.0474/0.0387	0.0486/0.0502/0.0371
user CityBlock	0.0141/0.0093/0.00984	0.0179/0.0177/0.0149	0.0179/0.0177/0.0149
user Tanimoto	0.0645/0.0407/**0.042**	0.0577/0.0473/0.0411	0.0565/0.0502/0.0409
item Loglikelihood	0.0141/0.00895/0.00957	0.0116/0.0167/0.0118	0.0104/0.0278/0.0128
item CityBlock	0.0122/0.00593/0.00648	0.00899/0.0126/0.00852	0.0079/0.022/0.00975
item Tanimoto	0.0176/0.00978/0.0109	0.013/0.0186/0.0131	0.0113/0.0307/0.0139

TABLE 4.4: IF-IDF performance (precision/recall/F-measure)

metric	2-recommendations	5-recommendations	10-recommendations
user Loglikelihood	0.061/0.0398/0.0409	0.051/0.0479/0.0389	0.0486/0.0502/0.0371
user CityBlock	0.0187/0.015/0.0147	0.0179/0.0177/0.0149	0.0179/0.0177/0.0149
user Tanimoto	0.0635/0.0381/0.0404	0.058/0.0475/**0.0416**	0.0572/0.0504/0.0412
item Loglikelihood	0.0141/0.00895/0.00957	0.0118/0.0168/0.012	0.0105/0.0279/0.0129
item CityBlock	0.0132/0.00687/0.00728	0.00954/0.0139/0.00917	0.00845/0.0242/0.0106
item Tanimoto	0.0183/0.0102/0.0114	0.0133/0.0189/0.0134	0.0114/0.0311/0.0141

In a comparison of the distance metric used to determine CF neighbourhoods, the user-based Tanimoto measure is better than others, in general. This measurement

calculates the number of solutions that two problems share, divided by the number of solutions that either problem can be solved by.

A clear contrast also exists between results for user- and item-based collaborative filtering approaches, with the former performing far better than the latter in all cases. It can be concluded from these results that it is far better to recommend new solutions to old problems, than to try and bring new problems to old solutions. In many ways this is an intuitive result, as it is far more likely that extant solutions will be immediately attempted when new research problems arise.

Finally, the effectiveness of the three idea relevance scores tested were considered. OCC, CON and IF-IDF are almost indistinguishable. OCC is slightly better than others. These results may indicate that simply counting idea-pair occurrences in the dataset is a sufficient basis to assess the significance of a solution to any given problem.

4.1.2 Discussion

This study demonstrated the plausibility of generating novel idea-candidates in an automated fashion. User-based collaborative filtering offered the best performance and, while different distance measures produced similar results, OCC provided a simple method to achieve the most effective performance. Nonetheless, there is scope for further research at each of the stages of the idea-generation process that has been presented. First, there is potential to improve the filtering of noun-phrases identified by POS tagging (based perhaps upon a more formalized information-theoretic approach to detecting 'semantically redundant' terms).

Second, the approach to classifying s-phrases and p-phrases remains relatively coarse, using a pattern matching approach based on trigger words. A further investigation of this processing stage would be of particular interest and numerous options seem viable. It is the aim, for example, to implement a supervised classification model to improve detection of *s-phrases* and *p-phrases*. The input features for this model could be generated from language models (Song & Croft, 1999),

lexical cohesion (Morris & Hirst, 1991) and linguistic grammar-based techniques (P. F. Brown et al., 1992), in addition to the statistical features already used. Training would need to be performed on ground truth annotations of scientific abstracts, but these could be collated in a crowd-sourced fashion by presenting abstracts to domain experts and allowing them to manually identify problem and solution term patterns within the text. The goal here would be to directly address some of the limitations with the current approach, such as the fact that *p-phrases* and *s-phrases* are overly dependent upon their context (for example, a p-phrase in one document might be an s-phrase in another).

Additional areas of interest lie not only in investigating other similarity measures from the collaborative filtering literature, but also in exploring other external indicators of a known-idea's relevance value. These might include the number of citations generated by the paper the idea appears in, or the impact factor of its parent publication, or indeed any of the host of methods that are used to assess the relevance of a paper within the scientific literature.

Finally, and perhaps of greatest importance, there is a good deal of room to extend the evaluation of the efficacy of the ideas generated by the system. Currently, a novel-idea candidate's merit based upon whether it occurs in (or is absent from) future literature was assessed. This neglects two factors: 1. the comprehensibility and interestingness of generated idea-pairs (a situation which can only be addressed by a programme of human evaluation of the system's outputs), and 2. any assessment of an idea's *inventiveness*. Currently, if a recommended idea does not appear in the test set, it is deemed as a false positive out of hand, whereas it may be the case that the idea is simply yet to be researched. This indicates that novel idea generation is a very open question. They are not necessarily wrong but a likelihood of being novel can be tagged along. It also reflects that challenge of defining novelty in a mathematical manner.

4.2 Idea detection using neural networks

4.2.1 Result

Quantitative Measurement.

To measure the performance of the models, precision and recall with respect to each category with different settings were presented. Due to the imbalanced dataset (O: 13247; B-P: 83; B-S: 81; I-P: 690; I-S: 561), the results of the category with less data were not promising. Table 4.5 and 4.6 showed the performance of the model CWE-NN using pretrained vectors based on ACL300 and GoogleNews respectively. In each table, results are compared according to different epoch settings. Both tables showed that more epochs did not help to improve the results. On the contrary, the result of the category I-S is getting worse when the epoch is set to 1000. The precision and recall scores for B-P are all 0, whereas the performance for detecting B-S is better. This is maybe due to the limited training dataset. For example, in test dataset, in the sentence *The goal of inverse reinforcement learning is to learn the reward function in Markov decision process from expert demonstrations*, the problem statement is *learn the reward function in Markov decision process from expert demonstrations*, where the word *learn* is the beginning of the problem. In training dataset, the word *learn* appeared 14 times, 11 out of which are labeled as *O* and labeled as B-S once - only labeled as B-P once.

Label	p/r 100 epo	p/r 200 epo	p/r 1000 epo
B-P	0/0	0/0	0/0
B-S	0.38/0.04	0.30/0.04	0.17/0.01
I-P	0.10/0.01	0.13/0.01	0.21/0.01
I-S	0.05/0	0.06/0.01	0.04/0
0	0.90/0.99	0.90/0.99	0.90/0.99
AVG	0.29/0.21	0.28/0.21	0.26/0.21

TABLE 4.5: Results generated by the model CWE-NN with respect to different training Epochs using pretrained word vectors on ACL300.

Label	p/r 100 epo	p/r 200 epo	p/r 1000 epo
B-P	0/0	0/0	0/0
B-S	0.23/0.04	0.38/0.04	0.30/0.04
I-P	0.20/0.01	0.16/0.01	0.21/0.01
I-S	0.15/0.01	0.07/0.01	0/0
0	0.90/0.99	0.90/0.99	0.90/0.99
AVG	0.30/0.21	0.30/0.21	0.28/0.21

TABLE 4.6: Results generated by the model CWE-NN with respect to different training Epochs using pretrained word vectors on GoogleNews.

Label	p/r ACL300	p/r GoogleNews	p/r Non-pretrained
B-P	0/0	0.10/0.02	0.06/0.02
B-S	0.16/0.09	0.29/0.09	0.20/0.10
I-P	0.18/0.02	0.15/0.02	0.23/0.04
I-S	0.30/0.10	0.34/0.07	0.24/0.08
0	0.91/0.98	0.91/0.99	0.91/0.98
AVG	0.31/0.24	0.36/0.24	0.33/0.24

TABLE 4.7: Results generated by the model Bi-LSTM-CRFs. ACL300 represents using pretrained word vectors based on ACL300; GoogleNews represents using pretrained word vectors based on GoogleNews; Non-pretrained means the embeddings are not coming from pretrained word2ved model.

Qualitative Analysis. An example generated by the Bi-LSTM-CRFs model is provided in Table 4.8.

Note that in the end, all the I-P words are concatenated together to form the Problem representation and so do the I-S. The <problem, solution> pairs are constructed using combination between the identified problem and solution sequences. Since this task is seen as NER, only the entity units (words) are evaluated, but not the P-S pairs. That is to say, <problem, solution> is essentially <B-P I-P ..., B-S I-S ...>. If there are more than one problem identified, each problem will be

Problem	Solution
fast Magnetic Resonance Imaging (MRI)	*S1:* reconstructing MR image ... in MRI
fast Magnetic Resonance Imaging (MRI)	*S2:* deep architecture dubbed ADMM-Net
fast Magnetic Resonance Imaging (MRI)	*S3:* a data flow graph ... MRI model
fast Magnetic Resonance Imaging (MRI)	*S4:* the net ...using L-BFGS algorithm.

TABLE 4.8: The annotations generated by the Bi-LSTM-CRFs model for the abstract in the paper published by Yang et al. (Yang et al., 2016)

paired with each solution and so does the solution. There is no semantic similarity check between the detected problem or solution sequences, thus, there might be redundancies.

To test if the model can detect a paper's contribution, the Bi-LSTM-CRFs were applied to analyse the abstract (Collins et al., 1999), which is shown below:

> Abstract This paper considers statistical parsing of Czech, which differs radically from English in at least two respects: (1) it is a highly inflected language, and (2) it has relatively free word order. These differences are likely to pose new problems for techniques that have been developed on English. We describe our experience in building on the parsing model of (Collins 97). Our final results – 80% dependency accuracy – represent good progress towards the 91% accuracy of the parser on English (Wall Street Journal) text.

The sequence *statistical parsing of Czech* was tagged as the problem, which is consistent with the main contribution according to the study in the paper (Qazvinian & Radev, 2008). The exact piece of exertion from that paper is shown below:

> This shows the importance of this fact, and that *Dependency Parsing of Czech* is one of the main contributions of this paper. Table 3 also shows the number of times each shared fact appears in P99-1065s citation summary sorted by occurrence.

4.2.2 Discussion

A problem for training and evaluation in experiments of this nature is that it is difficult to enforce consistent annotation rules due to the differing subjective perceptions of the annotators. For example, annotators quite often have differing opinions about which piece of information best expresses the novelty in an abstract. Some of the annotators highlighted the problem (solution) explanations stated in a clause[1], rather than selecting the actual problem (solution) names.

The performance of the model CWE-NN is not as good as Bi-LSTM-CRFs in that CWE-NN predicts most of the words as outside of the chunk (O). Note that in order to draw conclusion, statistical test is needed, which is out of this scope. We may conclude that the features of the problem or solution do not rely on individual words. For the same word, it can be both of the problem and solution depending on the context. Another experiment was performed by only using Bi-LSTM excluding CRF. The model only using Bi-LSTM predicted all the words as outside of the chunk (O). The combined model Bi-LSTM-CRFs showed its strength. Despite some literature showing that the Bi-Directional RNN-LSTM attained state of the art performance in both English and Hindi (Athavale et al., 2016). It was found out that the Bi-LSTM-CRFs model, tended to generate more than one solution for a given problem. It is likely that this was caused by the similar structures in the sentences containing the solution-like sequences in the training dataset. For example, the model would tag any sequence following the words *we propose, we introduce* as the solution. Although it is unable to rank the importance of the detected sequences, the model showed evidence that it can spot some interesting patterns that humans tend to ignore.

[1]The underlines sequence in the Figure 3.1 reflected this aspect.

4.3 Argumentative Zoning

4.3.1 Result

Results of classification for per category. Table. 4.9 to 4.14 show the classification performance of different settings. [2]

TABLE 4.9: Performances of sentence classification based on different models per category with feature size 100 on the corpus ACL+AZ (precision/recall/F-measure)

Method	AIM	CTR	BKG	BAS
AVGWVEC ACL+AZ 100	0.29/0.85/**0.43**	0.29/0.80/0.42	0.36/0.68/**0.47**	0.11/0.87/0.20
PARAVEC ACL+AZ 100	0.60/0.03/0.06	0.20/0.004/0.009	0.39/0.02/0.04	0.00/0.00/0.00
AVGWVEC BSWE 100	-	-	-	0.14/0.63/**0.23**
Cuewords	0.13/0.55/0.21	0.33/0.20/0.25	-	0.08/0.36/0.13
Teufel 2002	0.44/0.65/0.52	0.34/0.20/0.26	0.40/0.50/0.45	0.37/0.40/0.38
Baseline	0.30/0.07/0.11	0.31/0.12/0.17	0.32/0.17/0.22	0.15/0.05/0.07

TABLE 4.10: Performances of sentence classification based on different models per category with feature size 100 on the corpus ACL+AZ (precision/recall/F-measure) - continue Table 4.9

	TXT	OWN	OTH
AVGWVEC ACL+AZ 100	0.47/0.88/0.61	0.59/0.68/0.63	0.49/0.69/**0.57**
PARAVEC ACL+AZ 100	0.52/0.11/0.18	0.62/0.98/**0.76**	0.35/0.004/0.009
Teufel 2002	0.57/0.66/0.61	0.84/0.88/0.86	0.52/0.39/0.44
Baseline	0.56/0.15/0.23	0.78/0.90/0.83	0.47/0.42/0.44

TABLE 4.11: Performance of sentence classification for the model *AVGWVEC* based on different text corpus per category with feature size 100 (precision/recall/F-measure)

Method	AIM	CTR	BKG	BAS
AVGWVEC ACL+AZ 100	0.29/0.85/0.43	0.29/0.80/0.42	0.36/0.68/0.47	0.11/0.87/0.20
AVGWVEC MixedAbs 100	0.11/0.73/0.19	0.11/0.71/0.20	0.14/0.62/0.23	0.04/0.65/0.08
AVGWVEC Brown model 100	0.19/0.73/0.30	0.38/0.56/0.45	0.19/0.55/0.28	0.05/0.72/0.10
AVGWVEC BSWE 100	-	-	-	0.14/0.63/0.23
Cuewords	0.13/0.55/0.21	0.33/0.20/0.25	-	0.08/0.36/0.13
Teufel 2002	0.44/0.65/0.52	0.34/0.20/0.26	0.40/0.50/0.45	0.37/0.40/0.38
Baseline	0.30/0.07/0.11	0.31/0.12/0.17	0.32/0.17/0.22	0.15/0.05/0.07

[2]Note that it is not completely compatible with Teufel 2002 results, since the dataset is different due to the sentence concatenation in this study. But Teufel's reports could be a reference.

TABLE 4.12: Performance of sentence classification for the model *AVGWVEC* based on different text corpus per category with feature size 100 (precision/recall/F-measure) - continue Table 4.11

	TXT	OWN	OTH
AVGWVEC ACL+AZ 100	0.47/0.88/0.61	0.59/0.68/0.63	0.49/0.69/0.57
AVGWVEC MixedAbs 100	0.15/0.75/0.25	0.72/0.56/0.63	0.21/0.61/0.31
AVGWVEC Brown model 100	0.30/0.72/0.42	0.56/0.52/0.54	0.42/0.66/0.51
Teufel 2002	0.57/0.66/0.61	0.84/0.88/0.86	0.52/0.39/0.44
Baseline	0.56/0.15/0.23	0.78/0.90/0.83	0.47/0.42/0.44

TABLE 4.13: Performance of sentence classification with the model *AVGWVEC* based on different feature sizes on the corpus ACL+AZ (precision/recall/F-measure)

Method	AIM	CTR	BKG	BAS
AVGWVEC ACL+AZ 300	0.29/0.82/0.43	0.34/0.75/0.47	0.36/0.72/0.48	0.10/0.72/0.17
AVGWVEC ACL+AZ 100	0.29/0.85/0.43	0.29/0.80/0.42	0.36/0.68/0.47	0.11/0.87/0.20
Cuewords	0.13/0.55/0.21	0.33/0.20/0.25	-	0.08/0.36/0.13
Teufel 2002	0.44/0.65/0.52	0.34/0.20/0.26	0.40/0.50/0.45	0.37/0.40/0.38
Baseline	0.30/0.07/0.11	0.31/0.12/0.17	0.32/0.17/0.22	0.15/0.05/0.07

TABLE 4.14: Performance of sentence classification with the model *AVGWVEC* based on different feature sizes on the corpus ACL+AZ (precision/recall/F-measure) - continue Table 4.13

	TXT	OWN	OTH
AVGWVEC ACL+AZ 300	0.51/0.87/**0.64**	0.61/0.71/0.65	0.49/0.65/0.56
AVGWVEC ACL+AZ 100	0.47/0.88/0.61	0.59/0.68/0.63	0.49/0.69/0.57
Teufel 2002	0.57/0.66/0.61	0.84/0.88/0.86	0.52/0.39/0.44
Baseline	0.56/0.15/0.23	0.78/0.90/0.83	0.47/0.42/0.44

I examine the results from the following aspects: When the feature dimension is set to 100 and the training corpus is ACL, the results generated by different models (AVGWVEC,PARAVEC and AVGWVEC+BSWE for BAS category only) were compared. Looking at the F-measure, AVGWVEC performs better than PARAVEC, but PARAVEC gave a better precision results on several categories, such as AIM, CTR, TXT and OWN. From the results, we can see that PARAVEC model is not robust, for example, it performs badly for the category of BAS. For specific category classification, take the BAS category for example, the BSWE model outperforms others in terms of F-measure.

When the model is fixed to AVGWVEC and the training corpus is ACL, the feature size impact (300 and 100 dimensions) was investigated. From the F-measure, we can see that for some categories, 300-dimension features perform better than the 100-dimension ones, for example, CTR and BKG, but they are not as good as 100-dimension features for some categories, such as BAS.

When the model is set to AVGWVEC and the feature dimension is 100, the results computed from different training corpus (ACL+AZ, MixedAbs and Brown corpus) were compared. ACL+AZ outperforms others and brown corpus is better than MixedAbs for most of the categories, but brown corpus is not as good as MixedAbs for the category of OWN.

Finally, the results between word embeddings and the methods of cuewords, Teufel 2002 and baseline were compared. To evaluate word embeddings on AZ, the model AVGWVEC trained on ACL+AZ was focused. As we can see from the table. 4.11, the model of word embeddings is better than the method using cuewords matching. It also outperforms Teufel 2002 for most of the cases, except AIM, BAS and OWN. It won baseline for most of the categories, except OWN.

4.3.2 Discussion

From the classification results we can see that the type of word embeddings and the training corpus affect the AZ performance. As the simple model, $AVGWVEC$ performs better than others, which indicate averaging the word vectors in a sentence can capture the semantic property of statements. By training specific argumentation word embeddings, the performance can be improved, which can be seen from the case of detecting BAS status using $BSWE$ model.

There is no significant difference between the results generated by 300-dimension of features and 100 dimensions.

Training corpus affects the results. ACL+AZ outperforming others indicates that the topics of the training corpus are important factors in argumentative zoning. Although Brown corpus has more vocabularies, it doesn't win ACL+AZ. The possible reason could be that ACL+AZ contents are more relevant for this particular experiment.

In general, the classification performance of word embeddings is competitive in terms of F-measure and for most of the cases, it is better than the manually crafted features proposed by Teufel et al. (Teufel & Moens, 2002).

Methods	**Micro-F**	**Macro-F**	**Weigh-F**
ACL300	0.88	0.33	0.82
ACL100	0.87	0.33	0.82
Brown100	0.87	0.31	0.81
n-grams	0.60	0.87	-
"+dep+neg	0.76	0.90	-

TABLE 4.15: Performance of citation sentiment classification on dataset-basic.

4.4 Citation sentiment analysis

4.4.1 Result

The performances of citation sentiment classification on dataset-basic and dataset-implicit were shown in Table 4.15 and Table 4.16 respectively. The result of classifying positive and negative citations was shown in Table 4.17. To compare with the outcomes in the work of (Athar, 2011) (the dataset refers to 3.1.4), two records from their results were selected: the best one (based on features n-gram + dependencies + negation) and the baseline (based on 1-3 grams). From Table 4.15 we can see that the features extracted by (Athar, 2011) performed far better than word embeddings, in terms of macro-F (their best macro-F is 0.90, ours is 0.33). However, the higher micro-F score (the highest micro-F is 0.88, theirs is 0.78) and the weighted-F scores indicated that the method may achieve better performances if the evaluations are conducted on a balanced dataset. Among the embeddings, ACL-Embeddings performed better than Brown corpus in terms of macro-F and weighted-F measurements[3]. To compare the dimensionality of word embeddings, ACL300 gave a higher micro-F score than ACL100, but there is no difference between 300 and 100 dimensional ACL-embeddings when look at the macro-F and weighted-F scores.

Table 4.16 showed the sent2vec performance on classifying implicit citations with four categories: neutral, negative, positive and excluded. The method had a poor performance on detecting positive citations, but it was comparable [4] with both the baseline and sentence structure method (Athar & Teufel, 2012b) for the category

[3]I did not perform significant test for the comparison.
[4]the F score difference between each method is less than 0.05

Sentiment	**Baseline**	**Athar**	**ACL300**
NEU (F-score)	0.86	0.89	0.84
NEG (F-score)	0.14	0.62	0.44
POS (F-score)	0.40	0.55	0.27
Macro-F	0.47	0.69	0.44
Weighted-F	-	-	0.77
X vs NEU,NEG,POS (F-score)	0.990	0.996	0.997

TABLE 4.16: Performance of implicit citation sentiment classification.

Trained Corpus	**F-measure**
Brown100	0.85
ACL300	0.85
ACL100	0.85
PS-ACL300	0.85

TABLE 4.17: Performance of classifying positive and negative citations.

of neutral citations. With respect to classifying negative citations, the proposed method was not as good as sentence structure features but it outperformed the baseline. The results of classifying category X from the rest showed that the performances of the method and the sentence structure method are fairly equal.

Table 4.17 showed the results of classifying positive and negative citations using different word embeddings. The macro-F score 0.85 and the weighted-F score 0.86 proved that word2vec is effective on classifying positive and negative citations. However, unlike the outcomes in the paper of (Tang et al., 2014), where they concluded that sentiment specific word embeddings performed best, integrating polarity information did not improve the result in the experiment.

4.4.2 Discussion

In this experiment, the citation sentiment classification results based on word embeddings were reported. The binary classification results in Table 4.17 showed that word2vec is a promising tool for distinguishing positive and negative citations. From Table 4.17 we can see that there are no big differences among the scores generated by ACL100 and Brown100, despite they have different vocabulary sizes

(ACL100 has 14,325 words, Brown100 has 56,057 words). The polarity specific word embeddings did not show its strength in the task of binary classification. For the task of classifying implicit citations (Table 4.16), in general, sent2vec (macro-F 0.44) was comparable [5] with the baseline (macro-F 0.47) and it was effective for detecting neutral sentences (F-score 0.84) as well as separating X sentences from the rest (F-score 0.997), but it did not work well on distinguishing positive citations from the rest. For the overall classification (Table 4.15), however, the proposed method was not as good as hand-crafted features, such as n-grams and sentence structure features. It is reasonable to conclude from the experiment that word2vec technique can capture sentiment information in the citations to some extent, but hand-crafted features have better performance.

Since citation sentences are easily identified by the special citation numbers attaching to the citation sentence, this study put more efforts on citation sentiment analysis. Examining sentiment of an idea can potentially motivate researchers to create novel ideas, either by utilizing positive ideas to new business endeavors, or inventing new methods to solve existing problems. Simply examining one citation may create bias towards the idea. In the next section, a series of citations citing the same paper are analyzed, both internally (within one paper) and globally (cross multiple papers).

4.5 Idea flow analysis

4.5.1 Result

- ICSS classification.

  The results (F1 scores) of using SVM 5-cross validation scheme to classify the ICSSs into the 3 categories are shown in the Table 4.18.

  To evaluate the performance of $SLCM_V$ qualitatively, the ICSSs were clustered into three categories based on the $SLCM_V$. By manually analyzing the

[5] the F score difference between each method is less than 0.05

TABLE 4.18: Results (F1 scores) of ICSS classification using SVM.

Features	SVM
$count + ratio$	0.71
$SLCM_V$	**0.86**
$SLCM_V + count + ratio$	0.84

cluster results of the ICSSs shown in the Table 3.14, it was found out that the case w02-1011 (controversial) was miss grouped with the case A92-1018 (negative), but the rest of the three example were correctly clustered.

- GCSS clustering.

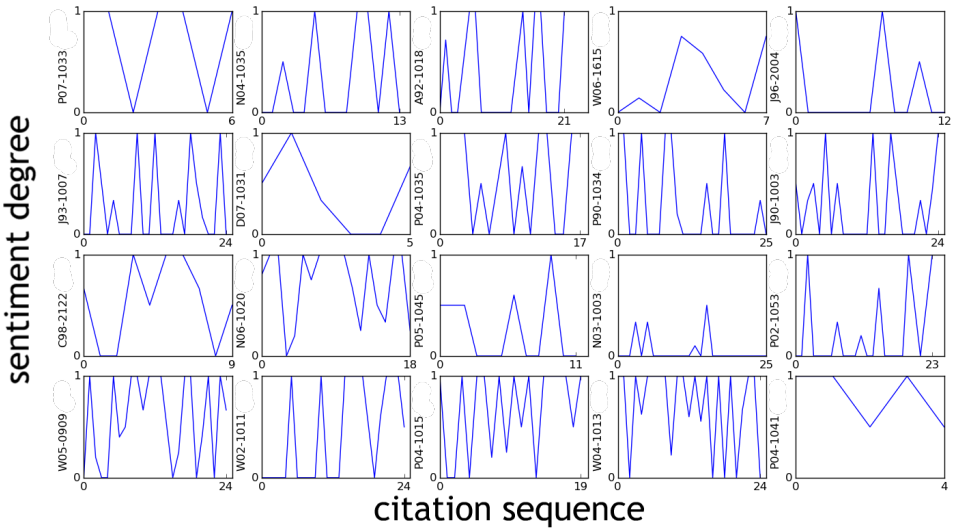


FIGURE 4.1: Plots of GCSSs, where the horizontal-axis represents the citations overtime (with respect to a specific cited paper.) and vertical-axis is the normalized sentiment value $pratio_l$ (The range of the value is 0-1, where 1 indicates strong positive and 0 strong negative). The label of each graph could be found on Dr.Athar's corpus page and the details of the citations information to construct the graph is demonstrated on the html file accordingly. For example: in the top-left graph, the cited paper with the label *p07-1033* got 6 citations, which are ordered by the citation published time. The bottom-right citation *p04-1041* has 4 citations. The citation normalized sentiment value is a single point according to the vertical-axis, but for visualizing the changed, the points were connected by lines, which form the curves on the graphs.

The clustering results computed by EDR method (namely: edr_{value}) for the 20 GCSSs are shown in the Figure 4.1[6]. The data are coming from the

[6]Using curves to represent the citation sentiment changes is to visualize the sequence turbulence. Some of them are also demonstrated in bar-chart 4.1.

samples that are demonstrated in the Table 4.19. Kmeans method was also used to cluster the 20 GCSSs into two categories based on the values $countcitations_g$, $countpapers_g$, $pratiopaper_g$,$pratio_g$, this method was called g_{value}. Since these two methods capture different characteristics in a sequence, the results based on these methods were not compared.

In order to illustrate the results more clearly, the graphs in Table 4.19 are labeled with numbers from 1 to 20 in an order from top left to bottom right, in a zigzag manner. From the clustering results shown in Table 4.19 we can see that there are disagreements between the $pratio_l$ and $countcitations_l$ based edr_{value} methods. For example, let's assume the cluster label 1 represents less fluctuated and 0 indicates more fluctuated, the method $pratio_l$ labeled the cases 3, 8 and 13 (A92-1018, P04-1035 and P05-1045) as 1, whereas the $countcitations_l$ based edr_{value} method labeled them as 0. The $countcitations_l$ based edr_{value} method is unable to compute the EDR score for the cases: 14, 20 and 4 (N03-1003, P04-1041 and W06-1615). When we look at the Table 3.14, the sequences 14 and 20 (N03-1003 and P04-1041) shared similar property that there are no opposite elements. This indicates The $countcitations_l$ based edr_{value} method is able to cluster some sequences successfully.

The $countcitations_l$ based edr_{value} method then clustered the rest of the sequences into two categories where one category only has one case : P07-1033. Based on an inspection of the data, it is likely that the algorithm sees the sentiment changes in this sequence are smooth. In comparison, the edr_{value} method using the $pratio_l$ as the element in the sequence to compute the EDR score showed its ability to compute the EDR scores for all the cases. However, this method is not good at discriminating sequences according to the ultimate purpose, which is to separate controversial citations from uncontroversial ones. The g_{value} cluster method took into account the sentiment scores that are accumulated from each citing papers, either by analyzing the citation counts, or the ratio of the positive citations.

TABLE 4.19: Results of GCSSs clustering using Kmeans (using the seven metrics derived by EDR). Each cell contains the cited paper ID and the cluster label generated by edr_{value} ($pratio_l$ / $countcitations_l$ /g_{value}), where $pratio_l$ is demonstrated in the equation 3.31, which is the fraction of positive citations among all the citations. $countcitations_l$ is explained in the equation 3.30. It is the difference between the number of positive citations and the negative ones. g_{value} considers sentiment scores that are accumulated from each citing papers. The 20 cells in this table are organized according to the sequences position in Figure 4.1.

P07-1033 (1/1/0)	N04-1035 (1/0/0)	A92-1018 (1/0/0)	W06-1615 (0/-/1)	J96-2004 (0/0/0)
J93-1007 (0/0/0)	D07-1031 (1/0/1)	P04-1035 (1/0/1)	P90-1034 (0/0/0)	J90-1003 (0/0/0)
C98-2122 (1/0/1)	N06-1020 (0/0/1)	P05-1045 (1/0/0)	N03-1003 (1/-/0)	P02-1053 (0/0/0)
W05-0909 (0/0/1)	W02-1011 (1/0/1)	P04-1015 (0/0/1)	W04-1013 (1/0/1)	P04-1041 (0/-/1)

4.5.2 Discussion

In this experiment, interesting approaches for measuring citation changes have been tested. Preliminary evidence showed that the proposed methods might be useful for information retrieval in a time series fashion. The method for classifying ICSSs is aimed at discovering the overall sentiment of a citing paper towards the cited paper. Several approaches were investigated, for example, the normalized and integrated positive and negative citation counts have been calculated and used as an intermediate results. The degree of citation sentiment changes could be related to the degree of texture roughness in an image, based on which, SLCM method was implemented. Supervised machine learning approach was adopted to classify the ICSSs into three categories: positive, negative and controversial. The best F1 score was achieved (F1 = 0.86) by using the $SLCM$ features. Different patterns of ICSSs can be distinguished by SLCM features. For example, the values of the four prototype for the sequence [1, 1, 1, 1, -1, -1, -1, -1] is (0.43, 0.14, 0.0, 0.43), whereas, the values of SLCM for the sequence [1, -1, 1, -1, 1, -1, 1, -1] is (0, 0.57, 0.43, 0). Although, the total number of 1s and -1s in these two sequences are the same, the SLCM unveils the differences from sentiment level changes perspective.

The method for clustering GCSSs focused on evaluating the paper's impact through time. Testing the citation sentiment changes over time is essentially testing the turbulence of a sequence. EDR method has been used in air turbulence evaluation, thus, it is reasonable to chose EDR as the empirical method. Based on EDR metrics computed from two types of sequences (the elements in the sequences are $pratio_l$ and $countcitations_l$ respectively), the 20 GCSSs are clustered into two categories, which are aiming at capturing the controversial characteristics in the sequence. The $countcitations_l$ based method showed some evidence to distinguish different patterns in the sequences, but the overall clustering performance is not robust enough to draw a conclusion. In order to overcome the data shortage, more GCSSs are needed. For example, 300 GCSSs would be good for further experiment. It is better that if the sequences are mixed with different level of turbulence samples.

The clusters generated by g_{value} are expected to reflect the sentiment level differences in the GCSSs. However, due to the limited dataset (20 GCSSs), a supervised machine learning model was not appropriate. Twenty samples are not enough for training a classification model using supervised machine learning methods. Considering the variations of the GCSSs forms, at least hundreds of samples are needed. On the other hand, different people have different opinion towards what is a smooth sequence and what is a fluctuated sequence. Therefore, collecting multiple labels for GCSSs from different people is necessary. The final label would be determined by the one that has most vote from the annotators. In this empirical study, the GCSSs were clustered into two classes that may indicate the overall sentiment level of all the citing papers towards one specific cited paper.

New schemes for computing the citation sequence scores need to be explored. The relations between the citations in the sequence maybe annotated as: attacking-to-supporting, supporting-to-attaching and none of the above, which could be detected by the prototyped shown in the Figure 2.5.

In order to use supervised machine learning techniques for classifying the citation sequences, a substantial amount of human annotated data is required, which will

involve manually analyzing the raw citations.

The possible application of this work can be used as the scientometrics for classifying the literatures according to the user's preferences (either favor the controversial papers to discover the research gap or the positive ones to innovate other ideas).

Chapter 5

Conclusion and Future work

This chapter is organized according to each method that have been introduced in 3.2. Each section in this chapter covers both conclusion and future work with respect to each method.

5.1 Idea mining and novel idea computation

5.1.1 Conclusion

This study presented a first approach for generating novel idea candidates from corpora of scientific text, that is decomposable into five distinct stages. Noun-phrases are extracted from the abstracts of scientific papers via POS tagging; a filtering process occurs to remove redundant concepts; the results set of phrases are subsequently categorized into problem and solution; co-occurring pairs are assigned a relevance score (based on number of co-occurrences, contribution to a problem's overall support or an idea frequency/inverse document frequency score); and finally a collaborative filtering algorithm generates new idea recommendations. This process illustrates the ability to transform of unstructured textual data into structured idea pairs, and the potential to manipulate that structure computationally to generate new idea candidates. The approach was evaluated using a

subset of publications from the journal *Science*, and both statistical and qualitative evaluations indicate strongly encouraging results, with an OCC relevance value combined with a (user-based) Tanimoto similarity measure offering the best performance. The hope is that in establishing this modular approach to automated idea generation, each stage may be honed by the broader research community to ultimately produce a system that has real utility to hypothesis generation.

To summarize, The main contributions of this study are listed bellow:

- The notion of an *idea* was formalized.
- A dataset constructed based on NLP techniques was obtained.
- Novel idea computation algorithm was implemented and tested.
- The computed ideas were released.

One challenge remaining is that a dataset with human labeled ground truth is highly difficult to obtain. It requires people to manually tag every line of text within a whole scientific publication corpora, trawling every paper in its entirety is almost intractable. Luckily we are afforded a potential solution in the form of *abstracts*, which are briefer pieces of text that distill the ideas held within a paper to a compact, annotatable blob of text. This potential solution of obtaining human annotated idea dataset is explored in the next experiment.

5.1.2 Future work

The representation of problem and solution could be further investigated. It might be useful to explore vector representations from other perspectives, such as applying attention Luong et al. (2015) mechanism on spotting the important words that can be considered as research problems or novel solutions. Regarding novel idea computation, other computational creativity methods could be explored, such as transformation and exploration. Human evaluation on the computed ideas could be the guide to select good ones.

5.2 Idea detection using neural networks

5.2.1 Conclusion

A human annotated dataset has been obtained containing 350 abstracts together with its scientific ideas (represented by the problem and solution sequences). Models (Bi-LSTM-CRFs and CWE-NN) for detecting the PSS automatically were trained and analyzed. The Bi-LSTM-CRFs model managed to tag some of the PSS correctly, but the overall performance needs to be improved. It is hoped in the future, the work may contribute to the information retrieval (IR) in such a way that based on the users problems, the IR system can retrieve the papers that contain the solutions that can potentially solve the problems.

The idea to detect PSS from abstract using deep learning technique is new. The model Bi-LSTM-CRFs can spot meaningful patterns in text that people have missed, which is very intriguing. This study is a direct application of neural network on PSS. The challenge in terms of using neural network is not domain specific. Thus, how to generalize the model to suit all writing styles needs further study.

The basic elements in our current conceptualization of an idea is based on problem and solution signals. These underpinned the explorations in this section and the section 5.1, both from machine learning and rule based perspective. But simply analyzing the elements of an idea may not be enough to understand the usage and the opinions towards the ideas from readers. In order to sense the idea from broader aspects, it is necessary to move from detecting phrases to analyzing sentences and zones, which contain ideas. By zones, it refer to chunks of text in a paper.

5.2.2 Future work

In the future, several aspects could be improved:

PSS ranking. To select the major problem (solution), how to rank all the identified sequences according to the similarity scores between the tagged sequence and the title is important. A Vector Space Model is a technique that could be used for measuring this similarity[1]. Alternatively, a supervised machine learning approach might solve this issue. In particular, an attention neural network (Bahdanau et al., 2014) for training the model to focus upon the most important parts of the information maybe useful.

Annotation tool for collecting more data. A web-interface for annotating the PSS were developed. In the future, it is possible to use crowd sourcing techniques to get the same abstracts annotated by many different people.

Many more annotations from the authors should be collected, and should approach as many authors of recent papers in this data set as feasible. The small number of author annotations that has been obtained are suggestive in terms of comparing the model predictions with the authors intentions. However, currently this sample size is too small to draw any objective conclusions.

Utilize the existing summarizers. The extractive summarizers, such as *text-summarizer* [2] select the most important sentences. It is likely that integrating part-of-speech tagging techniques into the summarizer would provide an effective mechanism for extracting the PSS.

Novel idea computation. Potential useful ideas can be discovered by analyzing the problem and solution sequences that are not seen together in one abstract. Using the similar method in the section 3.1.1, it is possible to accelerate the ideation process using collaborative filtering algorithm, where problem sequences are considered as users and solution sequences as the items to be recommended.

[1] https://github.com/mnielsen/VSM
[2] $http://textsummarization.net/text-summarizer$

5.3 Argumentative Zoning

5.3.1 Conclusion

In this study, different word embedding models on the task of argumentative zoning were compared. The results showed that word embeddings are effective on sentence classification from scientific papers. Word embeddings trained on a relevant corpus can capture the semantic features of statements and they are easier to be obtained than hand engineered features.

To improve the sentence classification for a specific category, integrating word specific embedding strategy helps. The size of the feature pool doesn't affect the results too much, nor does the vocabulary size. In comparison, the domain of the training corpus affects the classification performance.

The experiments conducted in this study indicates that idea mining is a challenging research area. Ideas remain nebulous concepts even within the constraints or the formalisms such as those proposed in the section 5.1. If there is a way to find non-natural language signals in text, which have special signals or warnings indicating ideas are nearby, then, these issues can be solved. In next section, one such potential indicator is examined, that is the *citation*.

5.3.2 Future work

The future work regarding this section is that the classification results should be utilized as the idea mining resources. For example, the sentences with the label AIM may contain research problems, since the aim of a paper is to solve a specific problem.

5.4 Citation sentiment analysis

5.4.1 Conclusion

The citation sentiment classification results based on word embeddings were reported. The binary classification results in Table 4.17 showed that word2vec is a promising tool for distinguishing positive and negative citations. From Table 4.17 we can see that there are no big differences among the scores generated by ACL100 and Brown100, despite they have different vocabulary sizes (ACL100 has 14,325 words, Brown100 has 56,057 words). The polarity specific word embeddings did not show its strength in the task of binary classification. For the task of classifying implicit citations (Table 4.16), in general, sent2vec (macro-F 0.44) was comparable [3] with the baseline (macro-F 0.47) and it was effective for detecting neutral sentences (F-score 0.84) as well as separating X sentences from the rest (F-score 0.997), but it did not work well on distinguishing positive citations from the rest. For the overall classification (Table 4.15), however, the proposed method was not as good as hand-crafted features, such as n-grams and sentence structure features. It is reasonable to conclude from the experiment that word2vec technique can capture sentiment information in the citations to some extent, but hand-crafted features have better performance.

5.4.2 Future work

Since citation sentences are easily identified by the special citation numbers attaching to the citation sentence, this section put more efforts on citation sentiment analysis. Examining sentiment of an idea can potentially motivate researchers to create novel ideas, either by utilizing positive ideas to new business endeavors, or inventing new methods to solve existing problems. The future work could be automatically writing literature review based on word2vec technique.

[3] the F score difference between each method is less than 0.05

5.5 Idea flow analysis

5.5.1 Conclusion

interesting approaches for measuring citation changes have been tested. Preliminary evidence showed that the proposed methods might be useful for information retrieval in a time series fashion. The method for classifying ICSSs is aimed at discovering the overall sentiment of a citing paper towards the cited paper. Several approaches were investigated, for example, the normalized and integrated positive and negative citation counts have been calculated and used as intermediate results. The degree of citation sentiment changes could be related to the degree of texture roughness in an image, based on which, SLCM method was implemented. Supervised machine learning approach was adopted to classify the ICSSs into three categories: positive, negative and controversial. The best F1 score was achieved (F1 = 0.86) by using the $SLCM$ features. Different patterns of ICSSs can be distinguished by SLCM features. For example, the values of the four prototype for the sequence [1, 1, 1, 1, -1, -1, -1, -1] is (0.43, 0.14, 0.0, 0.43), whereas, the values of SLCM for the sequence [1, -1, 1, -1, 1, -1, 1, -1] is (0, 0.57, 0.43, 0). Although, the total number of 1s and -1s in these two sequences are the same, the SLCM unveils the differences from sentiment level changes perspective.

The method for clustering GCSSs focused on evaluating the paper's impact through time. Testing the citation sentiment changes over time is essentially testing the turbulence of a sequence. EDR method has been used in air turbulence evaluation, thus, it is reasonable to chose EDR as the empirical method. Based on the seven EDR metrics computed from two types of sequences (the elements in the sequences are $pratio_l$ and $countcitations_l$ respectively), the 20 GCSSs are clustered into two categories, which are aiming at capturing the controversial characteristics in the sequence. The $countcitations_l$ based method showed some evidence to distinguish different patterns in the sequences, but the overall clustering performance is not robust enough to draw a conclusion. In order to overcome the data shortage, more

GCSSs are needed. For example, 300 GCSSs would be good for further experiment. It is better that if the sequences are mixed with different level of turbulence samples.

The clusters generated by g_{value} are expected to reflect the sentiment level differences in the GCSSs. However, due to the limited dataset (20 GCSSs), a supervised machine learning model was not appropriate. Twenty samples are not enough for training a classification model using supervised machine learning methods. Considering the variations of the GCSSs forms, at least hundreds of samples are needed. On the other hand, different people have different opinion towards what is a smooth sequence and what is a fluctuated sequence. Therefore, collecting multiple labels for GCSSs from different people is necessary. The final label would be determined by the one that has most vote from the annotators. In this empirical study, the GCSSs were clustered into two classes that may indicate the overall sentiment level of all the citing papers towards one specific cited paper.

5.5.2 Future work

New schemes for computing the citation sequence scores need to be explored. The relations between the citations in the sequence maybe annotated as: attacking-to-supporting, supporting-to-attaching and none of the above, which could be detected by the prototyped shown in Figure 2.5.

In order to use supervised machine learning techniques for classifying the citation sequences, a substantial amount of human annotated data is required, which will involve manually analyzing the raw citations.

The possible application of this work can be used as the scientometrics for classifying the literatures according to the user's preferences (either favor the controversial papers to discover the research gap or the positive ones to innovate other ideas).

References

Abu-Jbara, A., Ezra, J., & Radev, D. R. (2013). Purpose and polarity of citation: Towards nlp-based bibliometrics. In *Hlt-naacl* (pp. 596–606).

Ackoff, R. L. (1989). From data to wisdom. *Journal of Applied Systems Analysis*.

Agrawal, R., Imielinski, T., & Swami, A. N. (1993). *Mining association rules between sets of items in large databases, sigmod conference.*

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. , *31*(6), 3.

Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, *178*(1), 37–51.

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*(12), 2037–2041.

AIAA. (2012). Estimation of eddy dissipation rates from mesoscale model simulations. In *Aiaa aerospace sciences meeting including the new horizons forum and aerospace exposition.*

Alfonseca, E., & Man, S. (2002). An unsupervised method for general named entity recognition and automated concept discovery. In *1 st international conference on general wordnet.*

Ali, N. H., & Ibrahim, N. S. (2012). Porter stemming algorithm for semantic checking. In *Proceedings of 16th international conference on computer and information technology* (pp. 253–258).

Allan, J., Carbonell, J. G., Doddington, G., Yamron, J., & Yang, Y. (1998). Topic detection and tracking pilot study final report.

Altshuller, G. S. (1999). *The innovation algorithm: Triz, systematic innovation and technical creativity.* Technical Innovation Center, Inc.

Ambrosio, L., & Dal Maso, G. (1990). A general chain rule for distributional derivatives. *Proceedings of the American Mathematical Society*, *108*(3), 691-702.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., & Telgarsky, M. (2012). Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, *15*(1), 2773-2832.

Andrade, C. (2011). How to write a good abstract for a scientific paper or conference presentation. *Indian journal of psychiatry*, *53*(2), 172.

Angluin, D. (1992). Computational learning theory: survey and selected bibliography. In *Proceedings of the twenty-fourth annual acm symposium on theory of computing* (pp. 351–369).

Anwar A, A. (2010). Web information retrieval and search engines techniques. In *Al-satil journal* (p. 55-92).

Asur, S., Huberman, B., et al. (2010). Predicting the future with social media. In *Web intelligence and intelligent agent technology (wi-iat), 2010 ieee/wic/acm international conference on* (Vol. 1, pp. 492–499).

Athar, A. (2011). Sentiment analysis of citations using sentence structure-based features. In *Proceedings of the acl 2011 student session* (pp. 81–87).

Athar, A., & Teufel, S. (2012a). Detection of implicit citations for sentiment detection. In *Proceedings of the workshop on detecting structure in scholarly discourse* (pp. 18–26).

Athar, A., & Teufel, S. (2012b). Detection of implicit citations for sentiment detection. In *Proceedings of the workshop on detecting structure in scholarly*

discourse (pp. 18–26). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2391171.2391176

Athavale, V., Bharadwaj, S., Pamecha, M., Prabhu, A., & Shrivastava, M. (2016). Towards deep learning in hindi ner: An approach to tackle the labelled data scarcity.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ballatore, A., Bertolotto, M., & Wilson, D. C. (2014). An evaluative baseline for geo-semantic relatedness and similarity. *Geoinformatica*, *18*(4), 747-767.

Banerjee, S., & Pedersen, T. (2003). *The design, implementation, and use of the ngram statistics package*. Springer Berlin Heidelberg.

Banko, M., Etzioni, O., & Center, T. (2008). The tradeoffs between open and traditional relation extraction. In *Acl* (Vol. 8, pp. 28–36).

Barghout, L. (2015). *Spatial-taxon information granules as used in iterative fuzzy-decision-making for image segmentation*. Springer International Publishing.

Barlow, H. B. (2002). *Unsupervised learning*. John Wiley and Sons, Ltd.

Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2015). Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, *17*(4), 1-34.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, *3*, 1137–1155.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(1), 281-305.

Berry, M. (2004). *Survey of text mining :: clustering, classification, and retrieval*. Springer.

Besold, T. R., Khnberger, K. U., & Veale, T. (2015). Editorial: Computational creativity, concept invention, and general intelligence. *Journal of Artificial General Intelligence*, *6*(1), 1-4.

Besold, T. R., Schorlemmer, M., & Smaill, A. (2015). *Computational creativity research: Towards creative machines.* Atlantis Press.

Biemann, C., Böhm, K., Heyer, G., & Melz, R. (2004). Semantictalk: Software for visualizing brainstorming sessions and thematic concept trails on document collections. In *Knowledge discovery in databases: Pkdd 2004* (pp. 534–536). Springer.

Biemann, C., Böhm, K., Heyer, G., & Melz, R. (2006). Automatically building concept structures and displaying concept trails for the use in brainstorming sessions. In *Innovative internet community systems* (pp. 157–167). Springer.

Bikel, D. M., Miller, S., Schwartz, R., & Weischedel, R. (1998). Nymble: a high-performance learning name-finder. In *Conference on applied natural language processing* (pp. 194–201).

Bird, S., Dale, R., Dorr, B. J., Gibson, B. R., Joseph, M., Kan, M.-Y., ... Tan, Y. F. (2008). The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Lrec.*

Blair, D. C. (1979). Information retrieval, 2nd ed. c.j. van rijsbergen. london: Butterworths; 1979: 208 pp. *Journal of the Association for Information Science and Technology*, *30*(6), 374-375.

Blake, G. (1993). *Elements of technical writing.* Harcourt Brace Jovanovich.

Bleich, H. (1992). George boole, mathematical logic, and the modern computer. *M.d.computing Computers in Medical Practice*, *9*(3), 136.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Conference on computational learning theory* (p. 92-100).

Boden, M. A. (2004). *The creative mind: Myths and mechanisms.* Psychology Press.

Boden, M. A. (2009). Computer models of creativity. *AI Magazine*, *30*(3), 23.

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... Zhang, J. (2016). End to end learning for self-driving cars.

Bollegala, D., Maehara, T., & Kawarabayashi, K. I. (2015). Embedding semantic relations into word representations. In *International conference on artificial intelligence* (p. 1222-1228).

Bollmann, M., Bollmann, M., & Bollmann, M. (2013). Pos tagging for historical texts with sparse training data. In *Linguistic annotation workshop and interoperability with discourse* (p. 11-18).

Borthwick, A., Sterling, J., Agichtein, E., & Grishman, R. (1998). Description of the mene named entity system as used in muc-7.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. , 177-186.

Bowles, M. (2015). *4. penalized linear regression.* John Wiley and Sons, Inc.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the fourteenth conference on uncertainty in artificial intelligence* (pp. 43–52).

Breese, J. S., Heckerman, D., & Kadie, C. (2013). Empirical analysis of predictive algorithms for collaborative filtering. *Uncertainty in Artificial Intelligence*, *98*(7), 43-52.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on applied natural language processing* (pp. 152–155).

Brin, S. (1998). *Extracting patterns and relations from the world wide web.* Springer Berlin Heidelberg.

Brown, M. (2004). *Data mining techniques.* Wiley.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992, December). Class-based n-gram models of natural language. *Comput. Linguist.*, *18*(4), 467–479. Retrieved from http://dl.acm.org/citation.cfm?id=176313.176316

Bryant, R. E. (1986). Bryant, r.e.: Graph-based algorithms for boolean function manipulation. ieee trans. computers 35(8), 677-691. , *C-35*(8), 677-691.

Burges, C. J. C. (2008). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121-167.

Bybee, J. L., & Hopper, P. J. (2001). *Frequency and the emergence of linguistic structure* (Vol. 45). John Benjamins Publishing.

Cao, Z., Wei, F., Dong, L., Li, S., & Zhou, M. (2015). Ranking with recursive neural networks and its application to multi-document summarization. In *Twenty-ninth aaai conference on artificial intelligence.*

Caruana, R., Karampatziakis, N., & Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In *International conference on machine learning* (p. 96-103).

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Icml.*

Casasent, D. P., & Neiberg, L. M. (1995). Classifier and shift-invariant automatic target recognition neural networks. *Neural Networks the Official Journal of the International Neural Network Society*, *8*(78), 1117-1129.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection:a survey. *Acm Computing Surveys*, *41*(3), 1-58.

Chapelle, O., SchaLkopf, B., & Zien, A. (2006). Transductive support vector machines. In (p. 105-117).

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 321–357.

Chen, C. (2006). Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for information Science and Technology*, *57*(3), 359–377.

Chen, M., Hun, J., Yu, P. S., Ibm, T. J., & Ctr, W. R. (1996). Data mining: An overview from database perspective. In *Ieee transactions on knowledge and data engineering* (pp. 866–883).

Christensen, K., NÖrskov, S., Frederiksen, L., & Scholderer, J. (2017). In search of new product ideas: Identifying ideas in online communities by machine learning and text mining. *Creativity and Innovation Management*.

Cleveland, H. (1982). Information as a resource. *Futurist*, *34*(6), 34-39.

Cohen, J., Kempermann, J. H., & Zbaganu, G. (1998). *Comparisons of stochastic matrices with applications in information theory, statistics, economics and population*. Springer Science & Business Media.

Cohn, S. A. (1995). Radar measurements of turbulent eddy dissipation rate in the troposphere: A comparison of techniques. *Journal of Atmospheric and Oceanic Technology*, *12*(1), 85–95.

Collins, M., Ramshaw, L., Hajič, J., & Tillmann, C. (1999). A statistical parser for czech. In *Proceedings of the 37th annual meeting of the association for computational linguistics on computational linguistics* (pp. 505–512).

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (pp. 160–167).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537.

Colton, S., Pease, A., & Charnley, J. (2011). Computational creativity theory: The face and idea descriptive models. In *Proceedings of the second international conference on computational creativity* (pp. 90–95).

Connolly, T., Routhieaux, R. L., & Schneider, S. K. (1993). On the effectiveness of group brainstorming: Test of one underlying cognitive mechanism. *Small Group Research*, *24*(4), 490–503.

Cook, M., Colton, S., & Pease, A. (2012). Aesthetic considerations for automated platformer design. In *Aiide.*

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cortez, B., Carrera, B., Kim, Y. J., & Jung, J. Y. (2018). An architecture for emergency event prediction using lstm recurrent neural networks. *Expert Systems with Applications*, *97*, 315-324.

Culotta, A., & Sorensen, J. (2004). Dependency tree kernels for relation extraction. In *Proceedings of the 42nd annual meeting on association for computational linguistics* (p. 423).

Cuong, N. V., Ye, N., Lee, W. S., & Hai, L. C. (2014). Conditional random field with high-order dependencies for sequence labeling and segmentation. *Journal of Machine Learning Research*, *15*(1), 981-1009.

Das, D., & Martins, A. F. (2007). A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, *4*, 192–195.

Davenport, T. H., Prusak, L., & Prusak, L. (1997). *Working knowledge: How organizations manage what they know.* Harvard Business School Press.

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Icml '06 : Proceedings of the international conference on machine learning, new york, ny, usa* (pp. 233–240).

DebashisKushary. (1997). The em algorithm and extensions. *Technometrics*, *40*(3), 260-260.

Ding, W., & Chen, C. (2014). Dynamic topic detection and tracking: A comparison of hdp, c-word, and cocitation methods. *Journal of the Association for Information Science and Technology*.

Diplaros, A., Vlassis, N., & Gevers, T. (2007). A spatially constrained generative model and an em algorithm for image segmentation. *IEEE Transactions on Neural Networks*, *18*(3), 798-808.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*(7), 257-269.

Elliott, T., Plank, T., Zindler, A., White, W., & Bourdon, B. (2013). The application of clustering to earth science data: Progress and challenges. *Journal of Geophysical Research-Solid Earth*, *28*(1), 103-7.

Erilli, N. A., Yolcu, U., Eriolu, E., Alada, . H., & ner, Y. (2011). Determining the most proper number of cluster in fuzzy clustering by using artificial neural networks. *Expert Systems with Applications*, *38*(3), 2248-2252.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., & Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. *Eprint Arxiv*.

Fayyad, U., & Uthurusamy, R. (1996). *Data mining and knowledge discovery in databases*. ACM.

Fei, S., & Pereira, F. (2003). Shallow parsing with conditional random fields. In *Conference of the north american chapter of the association for computational linguistics on human language technology* (p. 134-141).

Foody, G. M., & Mathur, A. (2004). Toward intelligent training of supervised image classifications: directing training data acquisition for svm classification. *Remote Sensing of Environment*, *93*(1), 107-117.

Foster, J., zlem etinoglu, Wagner, J., Roux, J. L., Hogan, S., Nivre, J., ... Genabith, J. V. (2011). hardtoparse: Pos tagging and parsing the twitterverse. In

Analyzing microtext, papers from the 2011 aaai workshop, san francisco, california, usa, august.

Franke, J., Nakhaeizadeh, G., & Renz, I. (2003). *Text mining.* Physica-Verlag.

Fukuda, T. (1996). Data mining using two-dimensional optimized association rules. *Proc Acm Sigmod*, *25*(2), 13-23.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, *16*(10), 906-914.

Gallupe, R. B., Dennis, A. R., Cooper, W. H., Valacich, J. S., Bastianutti, L. M., & Nunamaker, J. F. (1992). Electronic brainstorming and group size. *Academy of Management Journal*, *35*(2), 350-369.

Garcia, E. M., Creus, C., Espaabonet, C., & Mrquez, L. (2017). Using word embeddings to enforce document-level lexical consistency in machine translation. *Prague Bulletin of Mathematical Linguistics*, *108*(1).

Ghazarian, S., & Nematbakhsh, M. A. (2015). Enhancing memory-based collaborative filtering for group recommender systems. *Expert Systems with Applications*, *42*(7), 3801-3812.

Glossary of terms. (1998). *Machine Learning*, *30*(2-3), 271-274.

Grefenstette, E., Dinu, G., Zhang, Y.-Z., Sadrzadeh, M., & Baroni, M. (2013). Multi-step regression learning for compositional distributional semantics. *arXiv preprint arXiv:1301.6939*.

Grosse, R. (1992). [does the self-driving farm machine have competition? potato and sugarbeet harvesting technology]. *Neue Landwirtschaft*.

Guo, S., et al. (2014). Analysis and evaluation of similarity metrics in collaborative filtering recommender system.

Guo, Y., Korhonen, A., & Poibeau, T. (2011). A weakly-supervised approach to argumentative zoning of scientific documents. *Proceedings of Emnlp*.

Han, J. (1996). Data mining techniques. In *Acm sigmod international conference on management of data, montreal, quebec, canada, june* (p. 545).

Haralick, R. M., Shanmugam, K., et al. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, *3*(6), 610–621.

Hare, V. C., & Milligan, B. (1984). Main idea identification: Instructional explanations in four basal reader series. *Journal of Literacy Research*, *16*(3), 189–204.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. In *The elements of statistical learning* (pp. 485–585). Springer.

Haykin, S. (1994). Neural networks: A comprehensive foundation. *Neural Networks A Comprehensive Foundation*, 71-80.

He, Y. D., & Kayaalp, M. (2006). A comparison of 13 tokenizers on medline. *Bethesda*.

Henry, N. L. (1974). Knowledge management: A new concern for public administration. *Public Administration Review*, *34*(3), 189.

Herlocker, J. L. (2015). Evaluating collaborative filtering recommender systems. In *The adaptive web* (p. 291-324).

Hiemstra, D., & Vries, A. P. D. (2017). Relating the new language models of information retrieval to the traditional retrieval models. *University of Twente Centre for Telematics and Information Technology*, *37*(4), 529-41.

Hildreth, P. M., & Kimble, C. (2004). *Knowledge networks: Innovation through communities of practice*. IGI Global.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527–1554.

Hinton, M. J., Geoffrey, & Rumelhart, D. (1986). Distributed representations.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hochreiter, S., & Schmidhuber, J. (2013). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.

Hollander, S. (1984). *Computer-assisted creativity and the policy process*. Thayer School of Engineering. Retrieved from http://books.google.com.my/books?id=qEiLtgAACAAJ

Hotho, A., Nrnberger, A., & Paass, G. (2005). A brief survey of text mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, *20*, 19-62.

Hsieh, Y. M., & Ma, W. Y. (2016). N-best rescoring for parsing based on dependency-based word embeddings. , *21*.

Huang, E. H., Socher, R., Manning, C. D., & Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Long papers-volume 1* (pp. 873–882).

Huang, L., Chen, H., Xun, W., & Chen, G. (2014). A fast algorithm for mining association rules. In *Ieee international conference on software engineering and service science* (p. 487-499).

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *Computer Science*.

Hui, B. (2001). Inferring semantics from collocation clusters to represent verbs and nouns.

Inakazu, K., Koakutsu, S., & Okamoto, T. (2016). An application of reccurent pulsed neural networks to time-series processing of autonomous mobile robot. *Ieej Transactions on Electronics Information and Systems*, *136*(7), 1017-1018.

Jackson, P., & Moulinier, I. (2007). *Natural language processing for online applications: Text retrieval, extraction and categorization* (Vol. 5). John Benjamins Publishing.

Jain, A. K. (1999). Data clustering: a review. *Acm Computing Surveys*, *31*(3), 264-323.

Jain, A. K., & Farrokhnia, F. (1991). *Unsupervised texture segmentation using gabor filters*. Elsevier Science Inc.

Jenks, G. F. (1967). The data model concept in statistical mapping. *International yearbook of cartography*, *7*(1), 186–190.

Jessop, J. L. (2002). Expanding our students' brainpower: Idea generation and critical thinking skills. *Antennas and Propagation Magazine, IEEE*, *44*(6), 140–144.

Jitendra, A. K., Cole, C. L., Hoppes, M. K., & Wilson, B. (1998). Effects of a direct instruction main idea summarization program and self-monitoring on reading comprehension of middle school students with learning disabilities. *Reading and Writing Quarterly: Overcoming Learning Difficulties*, *14*(4), 379–396.

Joachims, T. (1998). Making large-scale svm learning practical. In (p. 499-526).

Jochim, C., & Schütze, H. (2014). Improving citation polarity classification with product reviews. In *Acl (2)* (pp. 42–48).

Jones, K. S., Walker, S., & Robertson, S. E. (2000). *A probabilistic model of information retrieval: development and comparative experiments part 2*. Pergamon Press, Inc.

Kamp, H. (1981). A theory of truth and semantic representation. *Formal semantics-the essential readings*, 189–222.

Kaneda, Y., Ishihara, T., Yokokawa, M., Itakura, K., & Uno, A. (2003). Energy dissipation rate and energy spectrum in high resolution direct numerical simulations of turbulence in a periodic box. *Physics of Fluids*, *15*(2), L21–L24.

Ke, H., Chen, D., Li, X., Tang, Y., Shah, T., & Ranjan, R. (2018). Towards brain big data classification: Epileptic eeg identification with a lightweight vggnet on global mic. *IEEE Access*, *PP*(99), 1-1.

Kearns, M. J., & Vazirani, U. V. (1994). *An introduction to computational learning theory*. MIT Press,.

Kenter, T., Borisov, A., & de Rijke, M. (2016, August). Siamese cbow: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 941–951). Berlin, Germany: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P16-1089

Ketkar, N. (2014). Stochastic gradient descent. *Optimization*.

Kit, C., Xu, Z., & Webster, J. J. (2003). Integrating ngram model and case-based learning for chinese word segmentation. In *Sighan workshop on chinese language processing* (p. 160-163).

Kling, H. (1990). Get more out of group projects by using structured brainstorming. *QUALITY PROGRESS*, *23*(3), 136–136.

Koh, H. C., & Tan, G. (2005). Data mining applications in healthcare. *J Healthc Inf Manag*, *19*(2), 64-72.

Komninos, A., & Manandhar, S. (2016). Dependency based embeddings for sentence classification tasks. In *Conference of the north american chapter of the association for computational linguistics: Human language technologies* (p. 1490-1500).

Koopman, P. (1997). How to write an abstract. *Carnegie Mellon University. Retrieved May*, *31*, 2013.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *International conference on neural information processing systems* (p. 1097-1105).

Lafferty, J., McCallum, A., Pereira, F., et al. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, icml* (Vol. 1, pp. 282–289).

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In *Aaai* (Vol. 333, pp. 2267–2273).

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Larochelle, H., & Lauly, S. (2012). A neural autoregressive topic model. In *Advances in neural information processing systems* (pp. 2708–2716).

Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *Acm sigkdd international conference on knowledge discovery and data mining* (p. 16-22).

Lasic-Lazic, J., Seljan, S., & Stancic, H. (2000). Information retrieval techniques. In *2. carnetova korisnika konferencija cuc* (p. 2529-2532).

Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Lecun, Y. (n.d.). Lenet-5, convolutional neural networks.

Lenci, A., Montemagni, S., & Pirrelli, V. (2002). Acquiring and representing meaning: Theoretical and computational perspectives.

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and word2vec for text classification with semantic features. In *Cognitive informatics and cognitive computing (icci* cc), 2015 ieee 14th international conference on* (pp. 136–140).

Lin, C.-Y., & Hovy, E. (1997). Identifying topics by position. In *Proceedings of the fifth conference on applied natural language processing* (pp. 283–290).

Lin, D., & Wu, X. (2009). Phrase clustering for discriminative learning. In *Joint conference of the meeting of the acl and the international joint conference on natural language processing of the afnlp: Volume* (p. 1030-1038).

Lison, P., & Kutuzov, A. (2017). Redefining context windows for word embedding models: An experimental study. *arXiv preprint arXiv:1704.05781*.

Liu, B., & Zhang, L. (2012). *A survey of opinion mining and sentiment analysis*. Springer US.

Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *Computer Science*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1* (pp. 142–150).

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. of berkeley symposium on mathematical statistics and probability* (p. 281-297).

Madria, S. K., Bhowmick, S. S., Ng, W. K., & Lim, E. P. (1999). *Research issues in web data mining*. Springer Berlin Heidelberg.

Mäenpää, T. (2003). *The local binary pattern approach to texture analysis: extensions and applications*. Oulun yliopisto.

Manne, S., & Sameen Fatima, S. (2013). A feature terms based method for improving text summarization with supervised pos tagging. In *International symposium on information theory and its applications* (p. S676S677).

Manning, C. D., & Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Joint sigdat conference on*

empirical methods in natural language processing and very large corpora: Held in conjunction with the meeting of the association for computational linguistics (p. 63-70).

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Maron, M. E., & Kuhns, J. L. (1997). *On relevance, probabilistic indexing and information retrieval*. Morgan Kaufmann Publishers Inc.

Martin, S., Liermann, J., & Ney, H. (1998). Algorithms for bigram and trigram word clustering. In *European conference on speech communication and technology* (p. 19-37).

Mccallum, A., Freitag, D., & Pereira, F. C. N. (2000). Maximum entropy markov models for information extraction and segmentation. *Proc of Icml*, 591–598.

Mcdonald, R., Pereira, F., & Ribarov, K. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Conference on human language technology and empirical methods in natural language processing* (p. 523-530).

Merity, S., Murphy, T., & Curran, J. R. (2009). Accurate argumentative zoning with maximum entropy models. In *The workshop on text and citation analysis for scholarly digital libraries* (p. 19-26).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mitchell, J., & Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive science*, *34*(8), 1388–1429.

Mohanavalli, S., Jaisakthi, S. M., & Aravindan, C. (2011). Strategies for parallelizing kmeans data clustering algorithm. *Plos One*, *3*(3), e1828-e1828.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. MIT Press,.

Morris, J., & Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational linguistics*, *17*(1), 21–48.

Mntyl, M. V., Graziotin, D., & Kuutila, M. (2018). The evolution of sentiment analysisa review of research topics, venues, and top cited papers. *Computer Science Review*, *27*, 16-32.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *International conference on international conference on machine learning* (p. 807-814).

Nasukawa, T., & Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on knowledge capture* (pp. 70–77).

Nguyen, H. M., Cooper, E. W., & Kamei, K. (2011). Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, *3*(1), 4–21.

Nguyen, L. T., Wu, P., Chan, W., Peng, W., & Zhang, Y. (2012). Predicting collective sentiment dynamics from time-series social media. In *Proceedings of the first international workshop on issues of sentiment discovery and opinion mining* (p. 6).

Nijhuis, A. C. P. O., Yanovsky, F. J., Krasnov, O., & Unal, C. M. H. (2016). Assessment of the rain drop inertia effect for radar-based turbulence intensity retrievals. *International Journal of Microwave and Wireless Technologies*, *8*(6), 835-844.

Nivre, J., Hall, J., & Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. *In Proc. of LREC-2006*, 2216–2219.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(7), 971–987.

Oldale, A., Oldale, J., Reenen, J. V., & Campbell, M. (2004). *Collaborative filtering.* US.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607-9.

Osborn, A. F. (1953). Applied imagination.

Owen, S., Anil, R., Dunning, T., & Friedman, E. (2011). *Mahout in action.* Manning.

Paek, T., & Chickering, D. M. (2005). The markov assumption in spoken dialogue management. *Sigdial Workshop on Discourse and Dialogue*, 35–44.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345-1359.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, *2*(1-2), 1–135.

Park, Y., & Lee, S. (2011). How to design and utilize online customer center to support new product concept generation. *Expert Systems with Applications*, *38*(8).

Passos, A., Kumar, V., & Mccallum, A. (2014). Lexicon infused phrase embeddings for named entity resolution. *Computer Science*.

Pentheroudakis, J. E., Bradlee, D. G., & Knoll, S. S. (2006). *Tokenizer for a natural language processing system.* US.

Piatetsky-Shapiro, G. (1991). *Discovery, analysis, and presentation of strong rules.*

Pinel, F., Varshney, L. R., & Bhattacharjya, D. (2015). A culinary computational creativity system. In *Computational creativity research: Towards creative machines* (pp. 327–346). Springer.

Plank, B. (2009). Structural correspondence learning for parse disambiguation. In *Eacl 2009, conference of the european chapter of the association for computational linguistics, proceedings of the conference, march 30 - april3, 2009, athens, greece* (p. 37-45).

Pollack, M. E. (1995). Artificial intelligence a modern approach – a review. *AI Magazine*, *16*, 73-74.

Qazvinian, V., & Radev, D. R. (2008). Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd international conference on computational linguistics-volume 1* (pp. 689–696).

Radev, D. R., Jing, H., & Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 naacl-anlp workshop on automatic summarization* (pp. 21–30).

Raghavan, V., Bollmann, P., & Jung, G. S. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *Acm Transactions on Information Systems*, *7*(3), 205-229.

Rasmussen, E. (1999). In information retrieval: data structures and algorithms. *Lecture Notes in Computer Science*, *15*(5), 1211 - 1214.

Resnik, P. (2012). Semantic similarity based on corpus statistics and lexical taxonomy. , 11512–0.

Robertson, S. E., & Jones, K. S. (1976). Relevance weighting of search terms. *Journal of the Association for Information Science and Technology*, *27*(3), 129-146.

Rosenberg, C., Hebert, M., & Schneiderman, H. (2005). Semi-supervised self-training of object detection models. In *Ieee workshops on application of computer vision* (p. 29-36).

Rossiter, J. R., & Lilien, G. L. (1994). New brainstorming principles. *Australian Journal of Management*, *19*(1), 61-72.

Rowley, J. E., & Hartley, R. J. (2008). *Organizing knowledge : an introduction to managing access to information*. Ashgate.

Rumelhart, David, E., Hinton, Geoffrey, E., Williams, & Ronald, J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533-536.

Rumfitt, I. (1994). *Structures and categories for the representation of meaning. by timothy c. potts*. Cambridge University Press,.

Russell, S. J., & Norvig, P. (2010). Artificial intelligence: a modern approach. *Applied Mechanics and Materials*, *263*(5), 2829-2833.

Sadeghian, A., & Sharafat, A. R. (n.d.). Bag of words meets bags of popcorn.

Salton, G., & Harman, D. (2003). *Information retrieval*. John Wiley and Sons Ltd.

Salton, G., & Mcgill, M. J. (1983). *Introduction to modern information retrieval /*. McGraw-Hill.

Samuel, A. L. (1988). Some studies in machine learning using the game of checkers. i. In D. N. L. Levy (Ed.), *Computer games i* (pp. 335–365). New York, NY: Springer New York.

Sang, E. F. T. K., & Meulder, F. D. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. , *21*(08), 142-147.

Sang, T. K., & Erik, F. (2002). Introduction to the conll-2002 shared task: language-independent named entity recognition. *Computer Science*, 142–147.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. , 285–295.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International conference on new methods in language processing.*

Sekine, S. (1998). Nyu: Description of the japanese ne system used.

Shapiro, L. G., & Stockman, G. C. (n.d.). *Computer vision, 2001, 279-325.* New Jersey, Prentice-Hall, ISBN 0-13-030796-3.

Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., & Chanona-Hernndez, L. (2014). Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, *41*(3), 853-860.

Siebert, H., Lehmann, K., & Wendisch, M. (2006). Observations of small-scale turbulence and energy dissipation rates in the cloudy boundary layer. *Journal of the Atmospheric Sciences*, *63*(5), 1451-1466.

Sindhwani, V., & Melville, P. (2008). Document-word co-regularization for semi-supervised sentiment analysis. In *Data mining, 2008. icdm'08. eighth ieee international conference on* (pp. 1025–1030).

Soboroff, I., & Nicholas, C. (2000). Collaborative filtering and the generalized vector space model (poster session). In *International acm sigir conference on research and development in information retrieval* (p. 351-353).

Socher, R., Huang, E. H., Pennin, J., Manning, C. D., & Ng, A. Y. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems* (pp. 801–809).

Socher, R., Lin, C. C., Manning, C., & Ng, A. Y. (2011). Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (icml-11)* (pp. 129–136).

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 151–161).

Song, F., & Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the eighth international conference on information and knowledge management* (pp. 316–321).

Srivastava, N., Salakhutdinov, R. R., & Hinton, G. E. (2013). Modeling documents with deep boltzmann machines. *arXiv preprint arXiv:1309.6865*.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, *2009*, 4.

Sunkar, R. L. Y. J. G. (2015). Semantic similarity from natural language and ontology analysis. In (p. 254).

Takacs, A., Holland, L., Hueftle, R., Brown, B., & Holmes, A. (2005). Using in situ eddy dissipation rate (edr) observations for turbulence forecast verification..

Tambm publicado como CN1685371A, D. D. E. E. U. W., CN1685371B, & Bell, I. D. (n.d.). *Image recognition.*

Tan, A.-H., et al. (1999). Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases* (pp. 65–70).

Tang, D., Qin, B., & Liu, T. (2015). Deep learning for sentiment analysis: successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *5*(6), 292–303.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd annual meeting of the association for computational linguistics* (Vol. 1, pp. 1555–1565).

Taylor, J. W. (1961). *How to create new ideas.* Prentice-Hall.

Tenni, J., Lehtola, A., Bounsaythip, C., & Jaaranen, K. (2002). Machine learning of language translation rules. In *Ieee international conference on systems, man, and cybernetics, 1999. ieee smc '99 conference proceedings* (p. 171-177 vol.5).

Teufel, S. (2010). *The structure of scientific articles: Applications to citation indexing and summarization.* University of Chicago Press.

Teufel, S., Batchelor, C., & Batchelor, C. (2009). Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In *Conference on empirical methods in natural language processing: Volume* (p. 1493-1502).

Teufel, S., & Moens, M. (2002). Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, *28*(4), 409–445.

Teufel, S., et al. (2000). *Argumentative zoning: Information extraction from scientific text* (Unpublished doctoral dissertation). Citeseer.

Thorleuchter, D. (2008). Finding new technological ideas and inventions with text mining and technique philosophy. In *Data analysis, machine learning and applications - proceedings of the conference of the gesellschaft fr klassifikation e.v., albert-ludwigs-universitt freiburg, march* (p. 413-420).

Thorleuchter, D., den Poel, D. V., & Prinzie, A. (2010). A compared r and d-based and patent-based cross impact analysis for identifying relationships between technologies. *Technological Forecasting and Social Change*, *77*(7), 1037–1050.

Thorleuchter, D., Herberz, S., & Poel, D. V. D. (2011). *Mining social behavior ideas of przewalski horses.* Springer Berlin Heidelberg.

Thorleuchter, D., & Van den Poel, D. (2011a). Companies website optimising concerning consumer's searching for new products. In *Uncertainty reasoning and knowledge engineering (urke), 2011 international conference on* (Vol. 1).

Thorleuchter, D., & Van den Poel, D. (2011b). Semantic technology classificationa defence and security case study. In *Uncertainty reasoning and knowledge engineering (urke), 2011 international conference on* (Vol. 1, pp. 36–39).

Thorleuchter, D., & Van den Poel, D. (2012). Extraction of ideas from microsystems technology. In *Advances in computer science and information engineering* (pp. 563–568). Springer.

Thorleuchter, D., & Van den Poel, D. (2013). Web mining based extraction of problem solution ideas. *Expert Systems with Applications*, *40*(10), 3961–3969.

Thorleuchter, D., Van den Poel, D., & Prinzie, A. (2010a). Extracting consumers needs for new products-a web mining approach. In *Knowledge discovery and data mining, 2010. wkdd'10. third international conference on* (pp. 440–443).

Thorleuchter, D., Van den Poel, D., & Prinzie, A. (2010b). Mining ideas from textual information. *Expert Systems with Applications*, *37*(10), 7182–7188.

Tong, S., & Koller, D. (2000). Support vector machine active learning with applications to text classification. In *Seventeenth international conference on machine learning* (p. 999-1006).

Trampuš, M., & Mladenic, D. (2014). Constructing domain templates with concept hierarchy as background knowledge. *Information Technology And Control*, *43*(4).

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384–394).

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*, 141–188.

Ueda, T., & Togawa, F. (1993). *Receptive field neural network with shift-invariant pattern recognition.* US.

Ukil, A. (2002). Support vector machine. *Computer Science*, *1*(4), 1-28.

Uriartearcia, A. V. (2014). One-hot vector hybrid associative classifier for medical data classification. *Plos One*, *9*(4), e95715.

Van, M. G., & Bohte, S. (2017). Editorial: Artificial neural networks as models of neural information processing. *Frontiers in Computational Neuroscience*, *11*.

von Hardenberg, C., & Bérard, F. (2001). Bare-hand human-computer interaction. In *Proceedings of the 2001 workshop on perceptive user interfaces* (pp. 1–8).

New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/971478.971513 doi: 10.1145/971478.971513

Wallas, G. (1926). The art of thought.

Wang, C., Lu, J., & Zhang, G. (2007). Mining key information of web pages: A method and its application. *Expert Systems with Applications*, *33*(2), 425–433.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). Cnn-rnn: A unified framework for multi-label image classification. In *Computer vision and pattern recognition* (p. 2285-2294).

Wang, P., Xu, B., Xu, J., Tian, G., Liu, C.-L., & Hao, H. (2015). Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*.

White, C., Wood, K., & Jensen, D. (2012). From brainstorming to c-sketch to principles of historical innovators: Ideation techniques to enhance student creativity. *Journal of Stem Education Innovations and Research*, *13*(5), 12-25.

Widyantoro, D. H., Khodra, M. L., Riyanto, B., & Aziz, A. (2015). A multiclass-based classification strategy for rhetorical sentence categorization from scientific papers. *FormaMente: Rivista internazionale di ricerca sul futuro digitale*(3-2014), 223.

Willett, P. (2006). The porter stemming algorithm: then and now. *Program*, *40*(3), 219–223.

Williams, J., Cornman, L., Yee, J., Carson, S., Blackburn, G., & Craig, J. (2006). Nexrad detection of hazardous turbulence. In *44th aiaa aerospace sciences meeting and exhibit* (p. 76).

Williams, R. J., & Zipser, D. (1990). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*(2), 270-280.

Wong, S. K. M., & Raghavan, V. V. (1984). Vector space model of information retrieval: a reevaluation. In *International acm sigir conference on research and development in information retrieval* (p. 167-185).

Xiaojin, Z. (2007). Semi-supervised learning literature sur-vey. , *37*(1), 63-77.

Xue, B., Fu, C., & Shaobin, Z. (2014). A study on sentiment computing and classification of sina weibo with word2vec. In *Big data (bigdata congress), 2014 ieee international congress on* (pp. 358–363).

Xue, B., Fu, C., & Zhan, S. (2014). A study on sentiment computing and classification of sina weibo with word2vec. In *Ieee international congress on big data* (p. 358-363).

YANG, Q., & WU, X. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, *5*(04), -.

Yang, Y., Sun, J., Li, H., & Xu, Z. (2016). Deep admm-net for compressive sensing mri. *Adv. Neural Inf. Process. Syst*, *29*, 10–18.

Yessenalina, A., & Cardie, C. (2011). Compositional matrix-space models for sentiment analysis. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 172–182).

Yin, W., Kann, K., Yu, M., & Schtze, H. (2017). Comparative study of cnn and rnn for natural language processing.

Yin, Y., & Wang, D. (2017). The time dimension of science: Connecting the past to the future. *arXiv preprint arXiv:1704.04657*.

Yoon, J. (2012). Detecting weak signals for long-term business opportunities using text mining of web news. *Expert Systems with Applications*, *39*(16), 12543–12550.

Young, J. W. (1975). *A technique for producing ideas*. NTC Business Books.

Yu, C. N. J., & Joachims, T. (2009). Learning structural svms with latent variables. In *International conference on machine learning* (p. 1169-1176).

Yu, K., Schwaighofer, A., Tresp, V., Xu, X., & Kriegel, H. P. (2004). Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, *16*(1), 56-69.

Yu, L., Hermann, K. M., Blunsom, P., & Pulman, S. (2014). Deep learning for answer sentence selection. *Computer Science*.

Yu, Y., Lin, H., Meng, J., Wei, X., Guo, H., & Zhao, Z. (2017). Deep transfer learning for modality classification of medical images. *Information*, *8*(3), 91.

Zadeh, L. A. (1977). Pruf - a language for the representation of meaning in natural languages. In *International joint conference on artificial intelligence* (p. 918).

Zanzotto, F. M., Korkontzelos, I., Fallucchi, F., & Manandhar, S. (2010). Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 1263–1271).

Zarrella, G., & Marsh, A. (2016). Mitre at semeval-2016 task 6: Transfer learning for stance detection.

Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *CoRR*, *abs/1212.5701*. Retrieved from http://arxiv.org/abs/1212.5701

Zeleny, M. (1987). Management support systems: Towards integrated knowledge management. *Human Systems Management*, *7*(1), 59-70.

Zhang, D., Xu, H., Su, Z., & Xu, Y. (2015). Chinese comments sentiment classification based on word2vec and svm perf. *Expert Systems with Applications*, *42*(4), 1857–1863.

Zhao, H., Huang, C. N., Li, M., & Kudo, T. (2006). An improved chinese word segmentation system with conditional random field. *Proceedings of the Fifth Sighan Workshop on Chinese Language Processing*, 162–165.

Zou, W. Y., Socher, R., Cer, D., & Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1393–1398).