

FinNLP 2019

**The First Workshop on Financial Technology and
Natural Language Processing
in conjunction with IJCAI 2019**

Proceedings of the Workshop

August 12, 2019

Macao, China

Sponsor



科技部臺灣大學人工智慧研究中心

科技部人工智慧技術暨全幅健康照護聯合研究中心
MOST JOINT RESEARCH CENTER FOR AI TECHNOLOGY AND
ALL VISTA HEALTHCARE

Preface

The aim of the Workshop on Financial Technology and Natural Language Processing (FinNLP workshop) is to provide a forum for international participants to share knowledge on applying NLP to the FinTech domain. With the sharing of the researchers in the FinNLP workshop, the challenging problems of blending FinTech and NLP will be identified, the future research directions will be shaped, and the scope of this interdisciplinary research area will be broadened.

The 1st FinNLP workshop is held in Macao on August 12, 2019, in conjunction with the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019). The participants came from both academia and industry. Several novel tasks are presented, including business taxonomy construction, the rationale of trading, and contract disambiguation. Some new models are constructed for sales prediction, stock market prediction, and sentiment analysis. Tools and corpora related to FinTech and NLP are demonstrated. Furthermore, systems in the FinSBD shared task break the state-of-the-art of Sentence Boundary Detection in PDF Noisy Text in the Financial Domain.

We are immensely grateful to the Shared Task Organizers Sira Ferradans, Abderrahim Ait-Azzi, Guillaume Hubert, and Houda Bouamor. We also thank all of the Program Committee Members for their thorough review of the submissions. Besides, we would like to express our gratitude to MOST Joint Research Center of AI Technology and All Vista Healthcare for the financial support. Finally, many thanks to all participants for submitting their fine work and sharing their ideas. Without their efforts, this workshop could not be successful.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, Hsin-Hsi Chen

FinNLP 2019 Organizers

July 2019

Committees

Organizers

Chung-Chi Chen, National Taiwan University, Taiwan

Hen-Hsen Huang, National Chengchi University, Taiwan

Hiroya Takamura, National Institute of Advanced Industrial Science and Technology, Japan

Hsin-Hsi Chen, National Taiwan University, Taiwan

Shared Task Organizers

Sira Ferradans, Fortia Financial Solutions, France

Abderrahim Ait-Azzi, Fortia Financial Solutions, France

Guillaume Hubert, Fortia Financial Solutions, France

Houda Bouamor, Carnegie Mellon University, Qatar

Program Committee

Paulo Alves, Universidade Católica Portuguesa, Portugal

Avi Arampatzis, Democritus University of Thrace, Greece

Alexandra Balahur, European Commission's Joint Research Centre, Italy

Houda Bouamor, Carnegie Mellon University, Qatar

Paul Buitelaar, National University of Ireland Galway, Ireland

Damir Cavar, Indiana University, USA

Sunandan Chakraborty, Indiana University, USA

Brian Davis, Maynooth University, Ireland

Sira Ferradans, Fortia Financial Solutions, France

André Freitas, The University of Manchester, UK

Els Lefever, Ghent University, Belgium

Sheng Li, Google, USA

Nedim Lipka, Adobe Inc., USA

Heiner Stuckenschmidt, University of Mannheim, German

Ming-Feng Tsai, National Chengchi University, Taiwan

Chuan-Ju Wang, Academia Sinica, Taiwan

Wlodek Zadrozny, University of North Carolina in Charlotte, USA

Manel Zarrouk, National University of Ireland Galway, Ireland

Table of Contents

Business Taxonomy Construction Using Concept-Level Hierarchical Clustering Haodong Bai, Frank Xing, Erik Cambria and Win-Bin Huang.....	1
Towards Disambiguating Contracts for their Successful Execution - A Case from Finance Domain Preethu Rose Anish, Abhishek Sainani, Nitin Ramrakhiyani, Sachin Pawar, Girish K Palshikar and Smita Ghaisas.....	8
Rationale Classification for Educational Trading Platforms Annie Ying and Pablo Duboue.....	14
CoFiF: A Corpus of Financial Reports in French Language Tobias Daudert and Sina Ahmadi.....	21
Step-wise Refinement Classification Approach for Enterprise Legal Litigation Ying Mao, Xian Wang, Jianbo Tang and Changliang Li.....	27
CoSACT: A Collaborative Tool for Fine-Grained Sentiment Annotation and Consolidation of Text Tobias Daudert, Manel Zarrouk and Brian Davis.....	34
Financial Text Data Analytics Framework for Business Confidence Indices and Inter Industry Relations Hiroki Sakaji, Ryota Kuramoto, Hiroyasu Matsushima, Kiyoshi Izumi, Takashi Shimada and Keita Sunakawa.....	40
Learning to Learn Sales Prediction with Social Media Sentiment Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, Zihan Liu, Yan Xu, Cong Gao and Pascale Fung.....	47
Leveraging BERT to Improve the FEARS Index for Stock Forecasting Linyi Yang, Ruihai Dong, Tin Lok James Ng and Yang Xu.....	54
Economic Causal-Chain Search Using Text Mining Technology Kiyoshi Izumi and Hiroki Sakaji.....	61
Transformer-Based Capsule Network for Stock Movement Prediction Jintao Liu, Hongfei Lin, Xikai Liu, Bo Xu, Yuqi Ren, Yufeng Diao and Liang Yang.....	66
The FinSBD-2019 Shared Task: Sentence Boundary Detection in PDF Noisy Text in the Financial Domain Abderrahim Ait Azzi, Houda Bouamor and Sira Ferradans.....	74

AIG Investments.AI at the FinSBD Task: Sentence Boundary Detection through Sequence Labelling and BERT Fine-tuning	
Jinhua Du, Yan Huang and Karo Moilanen.....	81
aiai at FinSBD Task: Sentence Boundary Detection in Noisy Texts from Financial Documents Using Deep Attention Model	
Ke Tian and Zi Jun Peng.....	88
Pluto: A Deep Learning Based Watchdog for Anti Money Laundering	
Hao-Yuan Chen, Shang-Xuan Zou and Cheng-Lung Sung.....	93
From Creditworthiness to Trustworthiness with Alternative NLP/NLU Approaches	
Charles Crouspeyre, Eleonore Alesi and Karine Lespinasse.....	96
On a Chatbot Conducting a Virtual Dialogue in Financial Domain	
Boris Galitsky and Dmitry Ilvovsky.....	99
mhirano at the FinSBD Task: Pointwise Prediction Based on Multi-layer Perceptron for Sentence Boundary Detection	
Masanori Hirano, Hiroki Sakaji, Kiyoshi Izumi and Hiroyasu Matsushima.....	102
NUIG at the FinSBD Task: Sentence Boundary Detection for Noisy Financial PDFs in English and French	
Tobias Daudert and Sina Ahmadi.....	108
Sentence Boundary Detection of Financial Data with Domain Knowledge Enhancement and Bilingual Training	
Mingyu Wan, Rong Xiang, Emmanuele Chersoni, Natalia Klyueva, Kathleen Ahrens, Bin Miao, David Broadstock, Jian Kang, Amos Yung and Chu-Ren Huang.....	115
HITS-SBD at the FinSBD Task: Machine Learning vs. Rule-based Sentence Boundary Detection	
Mehwish Fatima and Mark-Christoph Mueller.....	122
AI_Blues at FinSBD Shared Task: CRF-based Sentence Boundary Detection in PDF Noisy Text in the Financial Domain	
Ditty Mathew and Chinnappa Guggilla.....	130

Conference Program

- 9:00-9:15 Opening Remarks
- 9:15-9:40 Business Taxonomy Construction Using Concept-Level Hierarchical Clustering
Haodong Bai, Frank Xing, Erik Cambria and Win-Bin Huang
- 9:40-10:05 Towards Disambiguating Contracts for their Successful Execution - A Case from Finance Domain
Preethu Rose Anish, Abhishek Sainani, Nitin Ramrakhiyani, Sachin Pawar, Girish K Palshikar and Smita Ghaisas
- 10:05-10:30 Rationale Classification for Educational Trading Platforms
Annie Ying and Pablo Duboue
- 10:30-11:00 Break
- 11:00-11:25 CoFiF: A Corpus of Financial Reports in French Language
Tobias Daudert and Sina Ahmadi
- 11:25-11:50 Step-wise Refinement Classification Approach for Enterprise Legal Litigation
Ying Mao, Xian Wang, Jianbo Tang and Changliang Li
- 11:50-12:15 CoSACT: A Collaborative Tool for Fine-Grained Sentiment Annotation and Consolidation of Text
Tobias Daudert, Manel Zarrouk and Brian Davis
- 12:15-13:30 Lunch
- 13:30-13:55 Financial Text Data Analytics Framework for Business Confidence Indices and Inter-Industry Relations
Hiroki Sakaji, Ryota Kuramoto, Hiroyasu Matsushima, Kiyoshi Izumi, Takashi Shimada and Keita Sunakawa
- 13:55-14:20 Learning to Learn Sales Prediction with Social Media Sentiment
Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, Zihan Liu, Yan Xu, Cong Gao and Pascale Fung
- 14:20-14:45 Leveraging BERT to Improve the FEARS Index for Stock Forecasting
Linyi Yang, Ruihai Dong, Tin Lok James Ng and Yang Xu
- 14:45-15:10 Economic Causal-Chain Search Using Text Mining Technology
Kiyoshi Izumi and Hiroki Sakaji
- 15:10-15:30 Break
- 15:30-15:55 Transformer-Based Capsule Network for Stock Movement Prediction
Jintao Liu, Hongfei Lin, Xikai Liu, Bo Xu, Yuqi Ren, Yufeng Diao and Liang Yang

- 15:55-16:15 **The FinSBD-2019 Shared Task: Sentence Boundary Detection in PDF Noisy Text in the Financial Domain**
Abderrahim Ait Azzi, Houda Bouamor and Sira Ferradans
- 16:15-16:25 **AIG Investments.AI at the FinSBD Task: Sentence Boundary Detection through Sequence Labelling and BERT Fine-tuning**
Jinhua Du, Yan Huang and Karo Moilanen
- 16:25-16:35 **aiai at FinSBD Task: Sentence Boundary Detection in Noisy Texts from Financial Documents Using Deep Attention Model**
Ke Tian and Zi Jun Peng
- 16:40-18:00 **Poster session**
- Pluto: A Deep Learning Based Watchdog for Anti Money Laundering**
Hao-Yuan Chen, Shang-Xuan Zou and Cheng-Lung Sung
- From Creditworthiness to Trustworthiness with Alternative NLP/NLU Approaches**
Charles Crouspeyre, Eleonore Alesi and Karine Lespinasse
- On a Chatbot Conducting a Virtual Dialogue in Financial Domain**
Boris Galitsky and Dmitry Ilvovsky
- mhirano at the FinSBD Task: Pointwise Prediction Based on Multi-layer Perceptron for Sentence Boundary Detection**
Masanori Hirano, Hiroki Sakaji, Kiyoshi Izumi and Hiroyasu Matsushima
- NUIG at the FinSBD Task: Sentence Boundary Detection for Noisy Financial PDFs in English and French**
Tobias Daudert and Sina Ahmadi
- Sentence Boundary Detection of Financial Data with Domain Knowledge Enhancement and Bilingual Training**
Mingyu Wan, Rong Xiang, Emmanuele Chersoni, Natalia Klyueva, Kathleen Ahrens, Bin Miao, David Broadstock, Jian Kang, Amos Yung and Chu-Ren Huang
- HITS-SBD at the FinSBD Task: Machine Learning vs. Rule-based Sentence Boundary Detection**
Mehwish Fatima and Mark-Christoph Mueller
- AI_Blues at FinSBD Shared Task: CRF-based Sentence Boundary Detection in PDF Noisy Text in the Financial Domain**
Ditty Mathew and Chinnappa Guggilla

Business Taxonomy Construction Using Concept-Level Hierarchical Clustering

Haodong Bai,[†] Frank Z. Xing,[‡] Erik Cambria,[‡] Win-Bin Huang^{†*}

[†]Department of Information Management, Peking University

[‡]School of Computer Science and Engineering, Nanyang Technological University

{hbai,huangwb}@pku.edu.cn, {zxing001,cambria}@ntu.edu.sg

Abstract

Business taxonomies are indispensable tools for investors to do equity research and make professional decisions. However, to identify the structure of industry sectors in an emerging market is challenging for two reasons. First, existing taxonomies are designed for mature markets, which may not be the appropriate classification for small companies with innovative business models. Second, emerging markets are fast-developing, thus the static business taxonomies cannot promptly reflect the new features. In this article, we propose a new method to construct business taxonomies automatically from the content of corporate annual reports. Extracted concepts are hierarchically clustered using greedy affinity propagation. Our method requires less supervision and is able to discover new terms. Experiments and evaluation on the Chinese National Equities Exchange and Quotations (NEEQ) market show several advantages of the business taxonomy we build. Our results provide an effective tool for understanding and investing in the new growth companies.

1 Introduction

Business taxonomies are important knowledge management tools for investment activities. When comparing different equity assets on the financial markets, investors tend to classify companies according to their main business sectors, market performances, and the products they manufacture. To discover companies with great potentials to grow across different industries, only those in the same industry sector will adopt similar criteria for downstream analysis, such as financial statement analysis, profit prediction, price-earnings valuation and more [Alford, 1992]. To this end, accurate classification of companies is crucial to successful investments. Consequently, governments and financial authorities, as well as big companies, have developed a large number of different business taxonomies, which are usually widely applicable, coarsely-grained and almost static. However, these features are not appropriate for small and startup companies.

These companies are often fast-growing, dynamically changing their business and focusing on a specific business. Therefore, traditional business taxonomies cannot reflect the whole landscape and emerging business. Beside the traditional business taxonomies, Chinese stock markets have yet another knowledge management tool called “concept stock (概念股)”. However, the concept labels are summarized by research teams and media, which means that they have already attracted much attention and over-represent blue chip stocks. Moreover, the concept labels are neither systematic nor hierarchical. One such influential label set is Tonghuashun’s “concept boards”¹. For small and startup companies, the current situation is that the valuation of such companies has to rely on concept labels transferred from the main domestic “A” shares markets, which do not appropriately describe small companies. The companies listed at the Chinese National Equities Exchange and Quotations (NEEQ)² are typical examples. Compared to those “A” share companies, the NEEQ listed companies rely even heavier on the inappropriate concept labels because there are no widely agreed market capitalization or enterprise multiple to them.

For the above-mentioned reasons, there is an urgent need for a more flexible business taxonomy to help with the investment decisions for small and new companies. The taxonomy can form benchmarks for thousands of different companies with innovative business models. Compared to the concept labels, a business taxonomy is not only helpful for investigating a specific company, but also beneficial to understand the relations between companies. There is already a large amount of studies on automatic taxonomy construction (ATC) for applications such as web search [Liu *et al.*, 2012], question answering and refinement [Sadikov *et al.*, 2010], advertising and recommendation systems, and knowledge organization [Zhang *et al.*, 2018]. However, few of them concerns business taxonomy construction. On the other hand, studies that leverage natural language processing (NLP) or text mining to support investment either improve the current existing taxonomy [Hoberg and Phillips, 2016] or express the industry structure using other mathematical tools [Xing *et al.*, 2019].

¹<http://q.10jqka.com.cn/gn/>

²The NEEQ is an over-the-counter (OTC) system for trading the shares of a public limited company that is not listed on either the Shenzhen or Shanghai stock exchanges, thus nicknamed “The New Third Board (新三板)”.

*Corresponding author: Win-Bin Huang

Unlike previous research, we propose a new method in this article that constructs a business taxonomy from scratch. The method extracts concept-level terms from the corporate annual reports, and computes the similarities between different terms. Based on the similarity matrix, the method recursively cluster terms into different strata.

Our *contributions* are tri-fold:

1. To the best of our knowledge, we pioneer the use of automatic taxonomy construction for the *business classification and investment purposes*. Using concept-level terms instead of keywords, the method needs a low level of supervision because we leverage linguistic knowledge and a statistical model to extract and compare terms. No seed terms or their relations are required.
2. We use positive and unlabeled learning (PU learning) to further mitigate the labor to tag indexing terms. The method thus shows its capability to identify fine-grained concepts and discover new terms from natural language.
3. We make the NEEQ annual reports dataset publicly available³, such that researchers could benchmark their taxonomy construction methods on it or follow up with other text mining tasks.

The remainder of this article is organized as follows: Section 2 elaborates related work from two thread of literature: the business classification systems and studies on automatic taxonomy construction; Section 3 provides an overview of the framework and introduce details of the algorithm; Section 4 presents experimental results; Section 4.2 evaluate the constructed taxonomy for the NEEQ market and carries out case studies; Finally, Section 5 concludes the study with future directions.

2 Related Work

2.1 Business Classification Systems

Business classification systems, or industry classification schemes, are fundamental tools for market research. According to a recent review [Phillips and Ormsby, 2016], companies are grouped and organized into categories by their similar manufacturing process, final products, and the target markets. Investors make use of the business classification systems for purposes such as benchmarking with flagship companies, discovering potential competitors, evaluating sales performances, and composing industry index. Mainstream business classification systems can be assorted into three classes depending on their developers and purposes: governmental statistical agencies develop the system for measuring economic activities, business information vendors develop the system for guiding investors, and academic researchers study the use of such system for accounting and finance. The most widely used examples are from business information producers, such as the Global Industry Classification Standard (GICS) and the Thomson Reuters Business Classification (TRBC), because they are integrated into the popular commercial databases. Early research [Bhojraj and Lee, 2002] also supports that

³The dataset is downloadable from the following link: <http://github.com/SenticNet/neeq-annual-reports/>.

the GICS accurately classifies the market. For this reason, some business classification systems used on the Chinese financial markets are adapted from GICS, such as the SWS classification standard⁴ and the official NEEQ classification guide⁵. However, many problems have been found when using these systems on the NEEQ market. First, designed using a top-down approach, these systems have unbalanced numbers of companies in the end-level of classes. To fit in a pre-defined structure, many classes contain companies with different businesses. Second, small companies are still at the early stage of exploring their business strategies. Therefore, it is common that one company’s business can span several domains in the system, while it can only be classified in a unique class. This causes the company’s absence in other classes. Last yet importantly, frequent revision of such systems is costly and would confuse investors.

Literature on using NLP and text mining for financial forecasting and investment activities is growing [Xing *et al.*, 2018]. Specific to business classification, Hoberg and Phillips built two systems using the 10-K corpus. The first one discovers competition relations between companies according to how similar are their product descriptions and constructs a company network [Hoberg and Phillips, 2010]. The second one first cluster companies with the text description of company products, then map the traditional business classification scheme to the newly constructed one [Hoberg and Phillips, 2016]. Both studies focused on improving the existing classification systems. Consequently, the details of a company’s business model are not revealed and classification results are still rather coarse. Taxonomies with more detailed information, for example on products [Aanen *et al.*, 2015], are not catered for the purpose of industry partition. In this research, we break the stereotype and take a fully data-driven approach for building the classification system based on the textual description of companies. The business-related concepts and terms are thus more detailed and information-rich.

2.2 Automatic Taxonomy Construction

A taxonomy is defined as a semantic hierarchy that organizes concepts by is-a relations [Wang *et al.*, 2017]. Since is-a relations are the most important relations in human cognitive structures, taxonomy construction from natural language is fundamental for ontology learning tasks. In common cases, ATC follows a pipeline of is-a relation extraction from natural language and induction of the taxonomy structure.

Relation extraction can be either pattern-based or statistical. One of the pioneer pattern-based research by Hearst [Hearst, 1992] proposed to use hand-crafted lexical patterns like “A is a B” and “A such as B” to discover is-a relations. More syntactic patterns are proposed by following research [Navigli *et al.*, 2011; Luu *et al.*, 2014], for example, “A, including B”, “A is a type/kind of B” etc. The performance can be improved by boosting over multiple such rules [Vivaldi *et al.*, 2001]. Pattern-based methods feature

⁴<http://www.swsindex.com/pdf/swhylfsm.pdf/>, Accessed on 2019-04-03.

⁵<http://www.neeq.com.cn/fenglei/hyfl.html/>, Accessed on 2019-04-03.

high precision but poor recall. This is because the exact match of such patterns has a low coverage over the relations contained in the corpus. This problem is more severe in our research because business descriptions usually do not contain explanatory clauses as above-mentioned in the linguistic patterns. Statistical model exams the relation between any two terms, i.e., first extract all the candidate terms, and build a model to predict what is the relation type or whether there exists an “is-a” relation between two terms. The term extraction step can be achieved with either supervised or unsupervised machine learning algorithms. In the former case, more label of true terms will be required and in the latter, only minimum effort is taken to threshold terms using TF-IDF, topic modeling (LDA) [Bakalov *et al.*, 2012], or TextRank model. For the relation predictive model, unsupervised methods leverage information such as co-occurrence frequency analysis, term subsumption [de Knijff *et al.*, 2013], cosine similarity based on bag-of-words, and word embedding similarities [Fu *et al.*, 2014] to discover taxonomic relations [Wang *et al.*, 2017]. Supervised methods require inductive reasoning over a set of known relations, which is more precise but rely heavily on the corpus as well as the seed relations [Zhang *et al.*, 2018]. In some cases, supervised methods have very poor recall. Obviously, there is a trade-off between precision and recall.

Induction of the taxonomy refers to the process of growing a graph-like structure based on the set of relations extracted from the previous step. The optimal taxonomy desires some features, such as no redundant edges and no loop of conceptual terms [Luu *et al.*, 2014]. The most important objective is the correctness of hypernym-hyponym relations: comparable terms should belong to the same level. Practically speaking, the business taxonomy should provide the necessary knowledge and business insights pertinent to the investment activities. To enable these, current approaches employ either clustering or algorithms that induct tree structure from a graph. Clustering methods assume that agglomerated terms share the same hypernym. By recursively choosing a representative term, hierarchical clustering can generate a layered tree structure [de Knijff *et al.*, 2013; Meijer *et al.*, 2014]. On the other hand, the term relations can be organized as a directed graph. Then the task becomes mining and pruning a tree structure out of the graph [Choi *et al.*, 2011]. In this research, we use a weakly supervised statistical method for relation extraction and greedy hierarchical affinity propagation (GHAP) to construct a new taxonomy, and relate companies to the leaf descendant layer.

3 Methodology

Our method can be divided into three phases: data preprocessing, concept-level taxonomy construction, and corporate categorization and labeling with the established taxonomy. Figure 1 provides an overview of the proposed method. Because the corpus we use is in Chinese, the data preprocessing phase consists of word segmentation and part-of-speech (POS) labeling of each Chinese word. We use the LTP-Cloud tools developed by HIT⁶ to complete this phase. The taxonomy construction phase utilizes a semi-supervised learning

⁶<http://www.ltp-cloud.com/>

Table 1: Concept-level features used to train a term extractor.

Name of features	Computing methods
Concept mutual information	$MI(t) = \sum_{i,j} p(i,j) \times \log[p(i,j)/(p(i)p(j))]$.
Right-side entropy	$RE(t) = \sum_i p(t,i) \times \log(p(t,i t))$.
Left-side entropy	$LE(t) = \sum_i p(i,t) \times \log(p(i,t t))$.
Concept TF	The overall term frequency in all the documents.
Concept IDF	The overall inverse document frequency in all the documents.
Followed-by word	Binary feature of whether the concept is followed by “industry (行业)” or “business scope (业务)”.
Following word	Binary feature of whether the concept is following “running (从事)”.
Industry TF	The concept frequency distribution in all the industry classes.
Industry IDF	The inverse document frequency distribution in all the industry classes.
Industry concept entropy	$IndE(t) = -\sum_i (TF_{t,i}/TF_t) \times \log(TF_{t,i}/TF_t)$.

classifier [du Plessis *et al.*, 2014] to reduce the amount of labor for tagging terms. After filtering out the concept term candidates, we obtain the final terms from the classifier. The similarity calculation is based on the idea of co-occurrence analysis from information science. Then GHAP takes the similarity matrix as an input to build a multi-layered structure of terms. The corporate categorization phase maps all the companies that contain the descendant-level terms to the taxonomy.

3.1 Concept Extraction and Term Similarity

One of the fundamental challenges in NLP is to model the semantic compositionality within phrases and multi-word expressions. Previous research [Cambria and White, 2014] suggests considering concepts to be the atomic units of meaning, which leads to more powerful expressiveness and more accurate results in downstream applications. Unlike ATC study which uses keywords [Liu *et al.*, 2012], we consider concept-level terms in our business taxonomy.

We observe that two types of templates together cover most of the concepts in the business domain, i.e. noun phrases and attributive phrases. For the first type, we mainly consider the noun-type POS tags in the “863 Chinese POS set”. Additionally, we include Chinese numerals⁷ and verbs, which are not morphologically identifiable to ensure a high recall. For the second type, we simultaneously consider the dependency parsing result. Those phrases that only contains dependency relation “ATT” (the attributive relation type in Chinese grammar) are selected to be concept term candidates.

The term candidates are represented with a concatenation of concept-level features as listed in Table 1 and similar word-level features. The features are designed to include both statistical and industry-related information based on the official NEEQ classification guide, because the distribution of term frequencies in texts of different industries is a crucial fact to the discriminative power of the term.

⁷Numerals appear in noun phrases such as “Third-party payment (第三方支付)”.

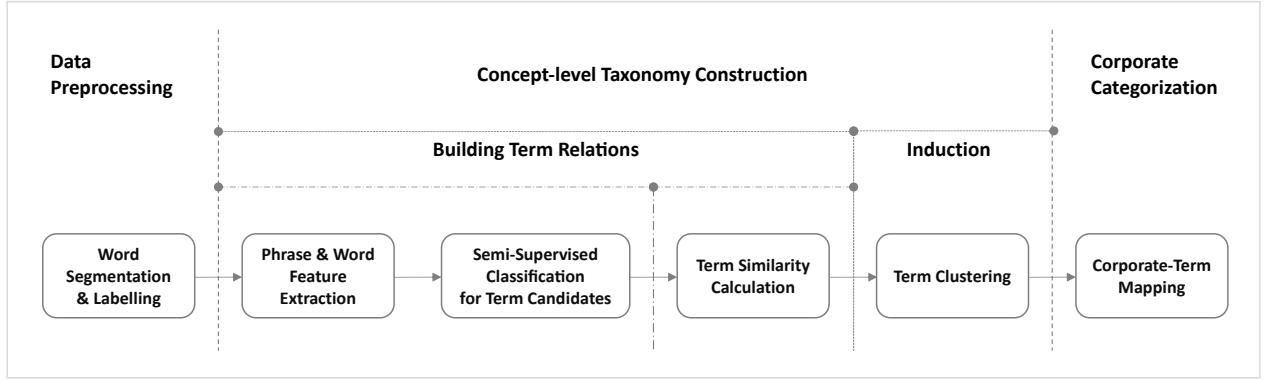


Figure 1: An overview of the proposed method, showing key techniques used in each module.

The semi-supervised classifier is built as a support vector machine (SVM) with probabilistic outputs under the framework of PU learning [du Plessis *et al.*, 2014]. PU learning is calibrated for real-world problems where labels of the negative cases are not accessible. Labels for positive cases are costly and hard to exhaust, so the majority of data remains unlabeled. Through the analysis of the empirical risk minimization problem of SVM, it is proved that PU learning is equivalent to a cost-sensitive classification where the cost ratio c_1/c_x is a function of class prior π and proportion of labeled sample η [du Plessis *et al.*, 2014]:

$$c_1/c_x = \frac{2\pi(1-\eta)}{\eta}. \quad (1)$$

We use the scikit-learn package to implement the cost-sensitive SVM with RBF kernel and estimate the probability parameters from the dataset. In experiments, we use the dual problem settings of PU learning, where only a small portion of negative cases are labeled. This is made possible by checking if the term candidate contains words from the stop-word list. We adapt a general stop-word list to the specific business domain by adding 106 domain-specific words to it. The added words include common words in the business domain such as “corporate (集团)”, “company (公司)” and action words such as “sales (销售)”, “profit (盈利)”, “leading (领先)”, “trend (趋势)” etc. After training with the negative labels, the classifier produces the real term set from the term candidates.

A term similarity is computed by integrating the comprising word-level similarities. To be more specific, we define the similarity of two words as the frequency of their co-occurrence divided by the harmonic mean of the frequencies of their occurrence in the documents respectively. That is

$$s(w_1, w_2) = \frac{2 \times dct(w_1 \cap w_2) \times dct(w_1) \times dct(w_2)}{dct(w_1) + dct(w_2)}, \quad (2)$$

where $dct(\cdot)$ denotes document counts. Then, we align corresponding words in two terms and use the average similarity of the best-match as the similarity between terms. Because this method is asymmetric, we define term similarity as the

average over two directions:

$$s(t_1 \rightarrow t_2) = \frac{\sum_{i \in t_1} \beta_i \max_{j \in t_2} s(i, j)}{len(t_1)} \quad (3)$$

$$s(t_1, t_2) = \frac{s(t_1 \rightarrow t_2) + s(t_2 \rightarrow t_1)}{2} \quad (4)$$

where i is word in term t_1 and j is word in term t_2 ; $len(t_1)$ denotes the length of t_1 . The weight for word i uses the TF-IDF information:

$$\beta_i = \log(ct(i)) \times \log\left(\frac{N}{dct(i)}\right). \quad (5)$$

where N is the total number of documents.

3.2 Taxonomy Induction

The term similarity matrix measures semantic relations between two given terms, where the target “is-a” relation is one of such. In order to construct a taxonomy, we computer a matrix of relations from the term similarity matrix by clustering, which preserve the strong relations while prune the others. We leverage greedy hierarchical affinity propagation (GHAP) [Xiao *et al.*, 2007], an exemplar-based clustering method to construct three layers of hypernym-hyponym relations. Compared to other clustering method, such as K-means, GMM or DBSCAN, GHAP has some advantages for taxonomy construction. *First*, the GHAP centroids are prototypical data points, which is important for the hypernym-hyponym relations. *Second*, GHAP does not need the number of clusters as a hyper-parameter input. *Third*, the clustering result of GHAP is insensitive to the initialization states. It is also worth mentioning that GHAP usually converges faster than HAP, which has to optimize a global loss function. The method is based on the concept of “message passing” between data points. For each layer, we iteratively compute a availability matrix $\mathbb{A}[\alpha_{ij}]_{n \times n}$ and a responsibility matrix

$\mathbb{R}[\rho_{ij}]_{n \times n}$ [Frey and Dueck, 2007], where

$$\alpha_{ii} = c_i + \sum_{k \neq i} \max(0, \rho_{ki}) \quad (6)$$

$$\alpha_{ij}^{i \neq j} = \min[0, c_j + \rho_{jj} + \sum_{k \notin \{i, j\}} \max(0, \rho_{ki})] \quad (7)$$

$$\rho_{ij} = s_{ij} - \max_{k \neq j} (\alpha_{ik} + s_{ik}), \quad (8)$$

i and j are taxonomic terms; c_j is the preference for choosing term j as an exemplar; n is the number of terms or exemplar terms in that layer. The binary exemplar vector is subsequently obtained as $\mathbf{e} = (\text{diag}(\mathbb{A}) + \text{diag}(\mathbb{R}) > 0)$. Each descendant term in this taxonomy further corresponds to a set of companies running similar business. A major difference of this taxonomy from traditional business classification systems is that one company can be mapped to multiple terms. This assumption is rational because in real-world cases, companies can span their business across several industry sectors.

4 Experiments and Evaluation

4.1 Data and Results

We crawled 21,739 annual reports for 10,375 listed companies from the NEEQ. The releasing time of these reports spans three years from 2014 to 2017. The original reports are in PDF format with relatively fixed discourse structure. We parse the files and extract texts from the section named “business model” using Tabula⁸. After manually cleaning the missing cases, we finally obtained 20,040 business model descriptions, summing up to 46.2 MB of textual data. According to the annual report standards, the descriptions cover the industry information, product and service, type of clients, key resource, sales model and components of income. Most of the descriptions comprise 100 to 1000 Chinese characters.

We obtained 64,460 concept-level term candidates from the corpus and labeled 7,078 of them as non-terms using the domain stop-word list. The cost-sensitive SVM classifier output 2,744 terms, which are clustered into 33 hypernoms (see Table 2). Our investigation shows that each hypernym governs no more than 20 sub-concept and 230 sub-sub-concept. Given the fact that the average term similarity equals 0.15, most of the clusters exhibit high intra-class similarity. We also observed a strong correlation between the numbers of sub- and sub-sub- concepts, which indicates the whole taxonomy is well-balanced.

To understand the branching structure within a hypernym, we showcase the structure of a relatively small ancestor class in the second row of Table 2 — “Education” (see Figure 2). There are four sub-concepts attached to this class: online training, professional training, education informatization, and smart education. Each sub-concept also has several hyponyms. Due to limited space we can not include all the education industry companies. Instead, we compare some popular NEEQ classification label and terms produced by our method.

⁸<http://tabula.technology/>

Table 2: Statistics of the first level hypernoms.

Hypernym	Intra-class similarity	No. of sub-concept	No. of sub-sub-concept	No. of companies
Healthcare 医疗诊断服务	0.40	2	17	72
Education 教育	0.37	4	15	137
Lighting 照明灯具	0.36	4	34	147
Game 游戏	0.34	3	33	156
Transportation & logistics 物流运输	0.33	3	22	206
Medical service & equipment 医疗器械制造与医疗服务	0.28	5	22	353
Ironmongery 金属零部件制造	0.27	4	26	208
Software & Hardware 第三方软硬件	0.27	4	51	525
Cement products 金属混凝土产品	0.27	3	9	34
Automobile 汽车	0.25	5	32	473
Electronics elements 电子原件制造	0.24	6	66	950
Telecoms 通信及通信设备	0.24	6	60	903
Building 建筑工程	0.24	7	59	433
Automation & robotics 自动化机器人	0.23	3	21	169
Information system & integration 信息系统集成服务	0.23	4	47	2416
Energy saving 节能环保	0.23	6	49	265
GIS service 地理信息服务	0.22	3	43	1601
IT infrastructure & maintenance IT基础设施与运维	0.22	4	32	252
Office appliance 日常办公用品	0.22	2	7	56
Digital media 互联网数字媒体	0.22	5	56	692
Clinical testing 临床试验检测	0.21	3	18	216
Smart houseware 智能家居	0.21	9	49	1086
Horticulture 园林工程	0.20	14	106	825
Mechanical equipment 机械设备制造	0.20	8	67	377
Chemicals 化工产品	0.19	6	35	274
Plastic products 塑料制品	0.19	12	59	395
Internet & online ads 互联网媒体广告	0.19	13	106	1097
Solar battery 太阳能电池	0.18	19	188	1699
E-commerce platforms 电商平台	0.17	8	53	1568
Financial services 金融服务	0.17	10	78	2673
Outsourcing consulting 工程咨询承包	0.17	10	79	4154
Natural bio-extract 天然植物提取物产品	0.16	18	125	1194
Phone gadgets 手机周边产品	0.16	20	223	8876

4.2 Qualitative Evaluation and Discussion

We benchmark the validity of our constructed business taxonomy with the official NEEQ classification guide via human evaluation. Generally, the descendant classes in the traditional business classification system are coarse. For example, many companies in the online education or training scope are classified as “Internet Software and Services”, which is apparently wilder-ranging; similarly, some companies are labeled as “General Customer Service”, which provides less information than the concept of “Online Training”. In fact, “Internet Software and Services” only reveals the means of conveying their product for online education companies. However, their customers, competitors, and market positioning are more comparable to traditional education companies, but are very different from internet software providers such as SAP or Tencent. In this sense, the traditional business classification system misleads investors by classifying companies with different business models together, providing inaccurate peers for pricing and research. In contrast, our method provides fine-grained concept-level terms. The mapping of companies are more balanced: each descendant term governs around ten companies in Table 2.

Another important aim of investment analysis is to discover new concepts and market trends. The new concepts often reflect how the industries will re-organize and develop

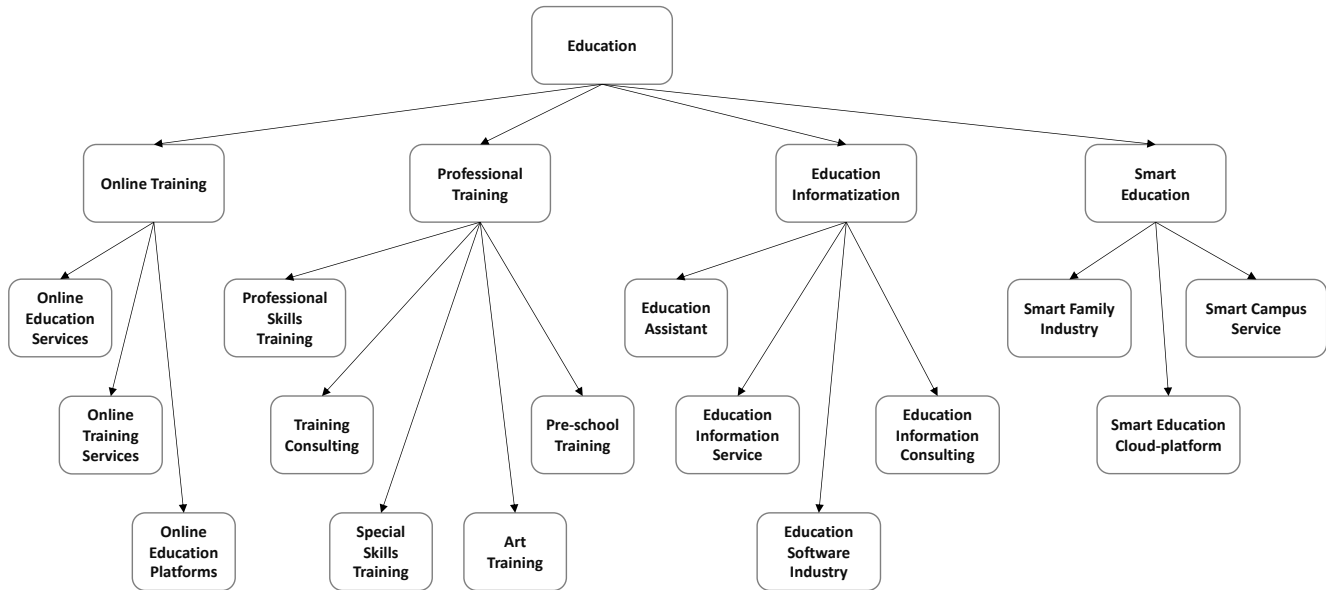


Figure 2: Three level classification system for the education industry.

in the future. However, the low frequency of update for traditional business classification systems tends to hide new business concepts. It is also challenging to find the appropriate position for new concepts. We notice that the business owners tend to advertise the hotspot concepts in their self-descriptions. Because our method is aware of the content of corporate annual reports, new concepts can be captured during taxonomy construction. For example, “online training” and “education informatization” are trendy concepts in the scope of education. Pre-school training is also increasingly popular in China, probably due to the Confucianist child-rearing ideas. These facts are not reflected in other business taxonomies for investment.

To summarize, our method allows *concrete terms* that would not appear in traditional business taxonomies to be displayed and facilitates the *discovery of new terms*. Therefore, the constructed taxonomy has some special advantages in investment activities compared to the static manually designed business classification systems, and can be a meaningful supplementary for the existing business classification systems.

5 Conclusion

In this article, we proposed a method to extract concept-level terms with weak and partial supervision and build a taxonomic structure of these terms using greedy hierarchical affinity propagation. The application of this method for business taxonomy construction is novel, for the reason that business texts have different linguistic features to represent “is-a” relations.

Our method is fast in both term similarity computing and taxonomy induction. Experiments on the Chinese NEEQ market show that the text-induced business taxonomy has several advantages over the traditional expert-crafted system, such as to display fine-grained concepts and discover trendy business concepts. The method provides a better tool for investment activities and industry research.

Of course, the constructed business taxonomy is not perfect. For instance, the “Phone gadgets” concept is giant and include too many companies. For this reason, the intra-class similarity is also the lowest for this class. These observations suggest that “Phone gadgets” can not be a good exemplar for the entire class and the class may be subject to further partition. Additionally, the semantic distances between hypernyms are at different scales: “Healthcare” and “Medical service and equipment” are small and related concepts that may be merged. Finally, the other relations between companies within the same set, e. g. supply chain relations, are not revealed. We will investigate how to improve the taxonomy with these relations in the future.

References

- [Aanen *et al.*, 2015] Steven S. Aanen, Damir Vandic, and Flavius Frasinca. Automated product taxonomy mapping in an e-commerce environment. *Expert Systems with Applications*, 42:1298–1313, 2015.
- [Alford, 1992] Andrew W. Alford. The effect of the set of comparable firms on the accuracy of the price-earnings

- valuation method. *Journal of Accounting Research*, 30(1):94–108, 1992.
- [Bakalov *et al.*, 2012] Anton Bakalov, Andrew McCallum, Hanna M. Wallach, and David M. Mimno. Topic models for taxonomies. In *ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 237–240, 2012.
- [Bhojraj and Lee, 2002] Sanjeev Bhojraj and Charles M. C. Lee. Who is my peer? a valuation-based approach to the selection of comparable firms. *Journal of Accounting Research*, 40(2):407–439, 2002.
- [Cambria and White, 2014] Erik Cambria and Bebo White. Jumping nlp curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2):48–57, 2014.
- [Choi *et al.*, 2011] Myung Jin Choi, Vincent Y. F. Tan, Animashree Anandkumar, and Alan S. Willsky. Learning latent tree graphical models. *Journal of Machine Learning Research*, 12:1771–1812, 2011.
- [de Knijff *et al.*, 2013] Jeroen de Knijff, Flavius Frasincar, and Frederik Hogenboom. Domain taxonomy learning from text: The subsumption method versus hierarchical clustering. *Data & Knowledge Engineering*, 83:54–69, 2013.
- [du Plessis *et al.*, 2014] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 703–711, 2014.
- [Frey and Dueck, 2007] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 305(5814):972–976, 2007.
- [Fu *et al.*, 2014] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. Learning semantic hierarchies via word embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1199–1209, 2014.
- [Hearst, 1992] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics (COLING)*, volume 2, pages 539–545, 1992.
- [Hoberg and Phillips, 2010] Gerard Hoberg and Gordon Phillips. Product synergies and competition in mergers and acquisitions: A text-based analysis. *The Review of Financial Studies*, 23(10):3773–3811, 2010.
- [Hoberg and Phillips, 2016] Gerard Hoberg and Gordon Phillips. Text-based network industries and endogenous product differentiation. *Journal of Political Economy*, 124(5):1423–1465, 2016.
- [Liu *et al.*, 2012] Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. Automatic taxonomy construction from keywords. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1433–1441, 2012.
- [Luu *et al.*, 2014] Anh Tuan Luu, Jung-Jae Kim, and See-Kiong Ng. Taxonomy construction using syntactic contextual evidence. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 810–819, 2014.
- [Meijer *et al.*, 2014] Kevin Meijer, Flavius Frasincar, and Frederik Hogenboom. A semantic approach for extracting domain taxonomies from text. *Decision Support Systems*, 62:78–93, 2014.
- [Navigli *et al.*, 2011] Roberto Navigli, Paola Velardi, and Stefano Faralli. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1872–1877, 2011.
- [Phillips and Ormsby, 2016] Ryan L. Phillips and Rita Ormsby. Industry classification schemes: An analysis and review. *Journal of Business & Finance Librarianship*, 21(1):1–25, 2016.
- [Sadikov *et al.*, 2010] Eldar Sadikov, Jayant Madhavan, Lu Wang, and Alon Halevy. Clustering query refinements by user intent. In *International World Wide Web Conference (WWW)*, pages 841–850, 2010.
- [Vivaldi *et al.*, 2001] Jordi Vivaldi, Llus Mrquez, and Horacio Rodriguez. Improving term extraction by system combination using boosting. In *European Conference on Machine Learning (ECML)*, pages 515–526, 2001.
- [Wang *et al.*, 2017] Chengyu Wang, Xiaofeng He, and Aoying Zhou. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1190–1203, 2017.
- [Xiao *et al.*, 2007] Jianxiong Xiao, Jingdong Wang, Ping Tan, and Long Quan. Joint affinity propagation for multiple view segmentation. In *International Conference on Computer Vision (ICCV)*, pages 1–7, 2007.
- [Xing *et al.*, 2018] Frank Z. Xing, Erik Cambria, and Roy E. Welsch. Natural language based financial forecasting: A survey. *Artificial Intelligence Review*, 50(1):49–73, 2018.
- [Xing *et al.*, 2019] Frank Z. Xing, Erik Cambria, and Roy E. Welsch. Growing semantic vines for robust asset allocation. *Knowledge-Based Systems*, 165:297–305, 2019.
- [Zhang *et al.*, 2018] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian M. Sadler, Michelle Vanni, and Jiawei Han. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2701–2709, 2018.

Towards Disambiguating Contracts for their Successful Execution - A Case from Finance Domain

Preethu Rose Anish*, Abhishek Sainani, Nitin Ramrakhiyani, Sachin Pawar,
Girish K Palshikar, Smita Ghaisas

TCS Research, India

{preethu.rose, a.sainani, nitin.ramrakhiyani, sachin7.p, gk.palshikar, smita.ghaisas}@tcs.com

Abstract

Contract management is key to financial services. Contracts lay down rules for doing business or present guidelines and recommendations for maximizing financial advantage of stakeholders in a given scenario. Contracts related to award of projects by companies to vendors, employment contracts, lease agreements, franchise agreement and even prenuptial agreements have significant financial implications. Making sense of contracts is an important step in achieving organizational goals such as building compliant systems, meeting delivery deadlines, avoiding heavy penalties and steering clear of expensive litigation. The complexity of “contracts language” however, makes it difficult to leverage the guidance they intend to offer. Contracts are written *ex ante*, based on forecasts rather than actual results, and may therefore contain ambiguous and incomplete guidance that can result in unintended violations. We address these problems by aiming to automate the disambiguation of contracts using a generalized architecture – R^3 to (1) Recognize important essential information present in contracts, (2) Reason over the information elements to identify their interrelations and uncover ambiguities and inconsistencies, and (3) Render the information in a visual format (for example, Message Sequence Charts) depicting different elements in contractual obligations.

1 Introduction

The digital transformation is well under way in the financial industry. Emerging business models, FinTechs, cryptocurrency, cyber threats, cyber security, data monetization, payments, greater collaboration and the shifting landscape of regulation and technology are all transforming this sector [Scardovi, 2017]. These transformations have accelerated the pace of business in the banking and financial sector leading to huge opportunities but also unpredictable risks.

A contract is a written agreement between two parties that details the terms of a transaction. Even though contracts are

fundamental in every business sector, their criticality is even more pronounced in the financial services for the reasons stated above. Contracts are vital to regulatory compliance and risk management, as they provide a deep insight into every aspect of an organization’s operation. This is especially significant in the financial sector, where, along with the accelerated pace of business, organizations face extremely stringent regulatory compliance requirements [Veerkar, 2018]. Failing to track compliance carries heavy financial risk. To cite just one example, during the calendar year ending 2018, close to fifteen banking firms and/or individuals were fined for breaching the principles of U.K.’s Financial Conduct Authority - FCA (FCA regulates financial firms providing services to consumers and maintains the integrity of the financial markets in the United Kingdom). The total amount of fine collected was close to £60 million. Clearly, a lack of attention to risk and compliance can cost a fortune to organizations and/or individuals.

An important first step towards achieving organizational goals of all participants involved in execution of contracts is to understand the contract documents comprehensively. However, the complexity of the documents makes it difficult to leverage the guidance they intend to offer. While the unintentional complexity in the documents may be due to the inherent peculiarities of natural language itself, the intentional complexity arises because the text aim to address and mitigate divergent and foreseeable, yet, unrealized scenarios in case they occur. Since contracts are written *ex ante*, based on forecasts rather than actual results, they may also contain ambiguous and incomplete guidance which can result in unintended violations and consequently, unfair penalties.

Contracts (similar taxonomically to regulations [Massey *et al.*, 2014]) contain obligations that must be fulfilled by the parties entering into a business agreement, the rights of the parties, permissions granted, exclusions to be made, and exceptions to the business rules. Additionally, contracts are bound by regulations prevalent in the countries where businesses are to operate, thereby inheriting the ambiguities in regulations. Of the different constructs (obligations, permissions, rights, exceptions, and exclusions) in a contract, it is easy to appreciate that obligations are the most demanding. Failure to fulfill contractual obligations can often entail punitive measures such as penalties and lengthy expensive litigations. The ongoing expensive contract breach case be-

*Contact Author

tween Apple and Qualcomm is a case in point [Reuters, 2017] wherein some communications between participants seem to have breached the non-compete clauses in the contract. While it is not yet clear if this breach was intentional, as noted earlier, due to the ambiguous and complex nature of the texts in contracts, or lack of complete information, unintentional breaches are also likely. How can we contribute to avoiding reducing such scenarios?

A number of people from customers' and vendors' organizations are involved in fulfilling contractual obligations. Their roles are mentioned in the documents while specifying the terms and conditions of an obligation. Contracts often also specify governance processes for fulfilling obligations. However, due to the large size of the documents, the convoluted Legalese-like language and the ambiguities, it is difficult to read contracts comprehensively and decipher who needs to do exactly what, while fulfilling an obligation. Disambiguating this complex text and presenting relevant information in a succinct, understandable form would therefore be useful to the participants. Our work on automating the disambiguation is motivated by the difficulties faced by people responsible for fulfilling contractual obligations.

We employ a generalized Recognize-Reason-Render (R3) architecture designed for disambiguating texts present in complex documents such as contracts and regulations. When we process contracts using the R3 architecture, the Recognize layer employs mechanisms to identify important elements such as obligations and corresponding actions and deadlines mentioned if any in the contract. The Reason layer establishes interrelations between the identified elements, for example, actor responsible for a given obligation or a trigger corresponding to an action to be taken for complying with the obligation. The Render layer constructs a user-friendly visual depiction in the form of Message Sequence Charts (MSC) to depict important actions, triggers, and actors involved in complying with contractual obligations.

The rest of the paper is organized as follows. In section II, we provide a brief overview of the R³ architecture. In section III, we present an illustration of disambiguation of a contract using the Recognize-Reason-Render components. Section IV is on related work and section V concludes the paper.

2 Overview of the R³ Architecture

The organizational goals such as avoiding penalties for non-compliance or ensuring no escalations in a project necessitate disambiguation of the documents that are meant to guide different stakeholders in fulfilling their responsibilities. For example, to make sure that a project does not run into escalations and penalties for unacceptable deliveries of products or services, we need to understand the obligations, rights, and the terms and conditions of service from a contract. With the aforesaid automation therefore, we must be able to address the following three aspects: (1) Recognize patterns that point to elements of interest aligned with the goal (for example - linguistic patterns characteristic of obligations and rights), (2) Apply reasoning that allows to derive meaning from the identified elements by determining their interrelations (such as trigger for the onset of an obligation), and (3) render the

output of reasoning in a way that can be understood by people responsible for executing contracts.

3 Applying R³ Architecture for Disambiguating Contracts

In this section, we present an illustration of disambiguation of a contract using the Recognize-Reason-Render components. Fulfilling contractual obligations so as to avoid heavy penalties and loss of credibility is a goal that motivates this disambiguation.

3.1 Recognize Layer

As stated earlier, contracts often outline governance processes for fulfillment of stated obligations. In our observation, a governance process consists of (i) multiple *actors* (which could be human users, organizations, designations or roles); and (ii) various *interactions* among actors, which are typically either *physical actions* (e.g., replace faulty parts) or *communication actions* involving exchange of information, instructions or control (e.g., approve, request, confirm, comply). In addition, a governance process must deal with external or internal (within the organization) *events* (e.g. termination) and *conditions* (e.g., increased costs, business interruption). The interactions are often ordered in specific ways, and may be associated with time expressions indicating durations, deadlines, frequency etc. Our goals are (i) to extract governance processes from a given contract; and (ii) express (or render) each governance process in a simple intuitive, visual form, which is easy for non-legal users to understand and implement in their workflows.

We categorize the information in contractual Obligation into TRIGGER, ACTION, ACTOR and TIMEX. We take a sample contract pertinent to financial construction and explain the categorization and the method that we have devised for extracting these pieces of information from the contracts document. Financing constructions is an important line of business for banks and housing finance companies. Financing a construction involves a legal contract or loan agreement, which typically includes governance processes for various eventualities, such as delays in completing the construction, or poor quality of construction. An example text in a legal contract¹ (available in public domain) containing description of a governance process is given below (entities are annotated).

[When [Contractor]_{Actor}'s Work is completed]_{Trigger}, [Contractor]_{Actor} shall [give written notice]_{Action} to [Owner]_{Actor} and/or [Bank]_{Actor}. [Owner]_{Actor} and/or [Bank]_{Actor} shall [make final inspection]_{Action} of [Contractor]_{Actor}'s work [within seven days]_{Timex} [after receipt of notice of completion]_{Trigger} and [upon acceptance thereof]_{Trigger} by [Owner]_{Actor} and [Bank]_{Actor}, [payment shall be made of the remaining balance due]_{Action}. Such [acceptance shall not be unreasonably withheld]_{Action} and

¹From sample contract at http://www.basnettdbr.com/pdfs/ConstCont_101117.pdf

[in the event [Owner]_{Actor} and/or [Bank]_{Actor} refuses to accept]_{Trigger}, [Owner]_{Actor} and/or [Bank]_{Actor} as the case may be shall [within said seven day period]_{Timex} [notify [Contractor]_{Actor} in writing of such refusal]_{Action} and shall [specify the reasons therefore]_{Action}. [Contractor]_{Actor} shall [within the next ten days]_{Timex} [take appropriate steps to remedy any deficiencies set forth as reasons for refusal]_{Action} and [upon completion thereof to the satisfaction of [Owner]_{Actor} and/or [Bank]_{Actor}]_{Trigger} shall [be entitled to prompt payment of any remaining balance due]_{Trigger}.

Using the above example text, we first provide a definition of each of the element we extract from contracts.

- **OBLIGATION (OBLG)** – a statement in the legal contract requiring a specific commitment from a user, which she must be made aware of. A governance process is expressed through OBLG statements.
- **TRIGGER** - a part of an OBLG statement containing a condition or event (e.g., an alert, an abnormal situation) that must be monitored by a user. Example: `When contractor's work is completed.`
- **ACTION** - a part of an OBLG statement containing a concrete action that must be taken by a user. Example: `give written notice.` ACTION is almost always paired with the corresponding trigger. Example, `[When contractor's work is completed]TRIGGER, [Contractor shall give written notice to Owner and Bank]ACTION.`
- **ACTOR** - typically, people, roles or organizations. Example: `Bank, Contractor, Owner.`
- **TIMEX** - time expressions (e.g., deadlines) associated with actions and triggers. Example: `within seven days`

We now outline our method for extracting the elements of a governance process from the text within a legal contract.

OBGL Classification: We use Multinomial Naïve Bayes as the classifier to predict the class label (OBGL or NOT_OBGL) for each sentence in the given contract. For identifying obligations, the classifier obtained a Precision, Recall and F-score of 91.8%, 91% and 91.3% respectively.

Trigger Extraction: Next, we detect and extract the text fragment corresponding to TRIGGER in the given OBLG sentence *S*, using rule-based *information extraction* (IE). First, we create an enriched representation of *S*, by adding information like POS tags, phrase structures, dependency relations, semantic roles and named entities. We then write simple regex patterns on this enriched text to extract triggers, which makes the patterns much more general and does not depend too much on detailed structure of the trigger text fragment. For example, triggers after receipt of notice of completion, or upon completion thereof to the satisfaction of, are reliably indicated by *cue words* such as *when* or *upon* and the rest of the trigger bears a specific dependency relation with this cue word.

For extracting triggers from the obligation statements, we obtained a precision of 86.5% and a recall of 80%.

Action Extraction: Next, we detect and extract the text fragment corresponding to ACTION in the given OBLG sentence *S*, using rule-based IE, similar to that used for TRIGGER extraction. For ACTION extraction, we obtained a Precision of 76.1

Actor Extraction: For ACTOR extraction, we follow the actor extraction algorithm in [Patil *et al.*, 2018]. The approach not only identifies canonical mentions of various actors but also their aliases mentioned as pronouns and generic noun phrases (NP). The algorithm utilizes WordNet hypernym structure to identify actor mentions. Then it uses first order logic rules containing linguistic knowledge to infer aliases in Markov Logic Networks framework.

Timex Extraction: We use the HeidelbergTime tool [Strotgen and Gertz, 2010] to extract time expressions from OBGL statements. It defines a temporal expression as a tuple of three elements: time expression as it occurs in the textual document, type of expression, and its value in a normalized form. To extract a temporal expression, this tool uses hand-crafted rules such that each rule consists of rule for identifying the time expression, information about the type of expression, and a function to normalize the identified time value. It further uses post-processing steps to resolve underspecified values and remove invalid temporal expressions.

3.2 Reason Layer

Having recognized and extracted the governance process elements described in sub-section 2.1, we employ the Reason layer to establish associations between them. We extract the governance process and map it to a Message Sequence Chart (MSC) [Mauw, 1997]. Every actor gets a separate timeline in the MSC. Actors are associated with the actions they initiate or receive. In the example contract, `Contractor` is the initiator of the action `deliver a written notice` and `Owner` and `Bank` are recipients (beneficiaries) of this action.

Since we extract the governance process and model it as an MSC, we can potentially identify ambiguities. In the example process, the phrase `prompt payment` is ambiguous because although `prompt` is a time indicator, it cannot be associated with any concrete deadline for `payment`. Use of the word `unreasonably` yields a similar ambiguity. Such ambiguities can be brought to the notice of the users, who can subsequently clarify them. Apart from ambiguities, one can also identify *incompleteness* issues with the governance process in the contract. For example, the example text does not include the governance process that corresponds to the *time-out* event; e.g., what action is to be taken if the `Owner` does not notify the `Contractor` of the refusal to accept the work, within ten days. This kind of incompleteness can potentially lead to lack of action on part of the participants (`Owner` and `Contractor` in this case). This kind of incompleteness can potentially lead to lack of action on part of the participants. The lack of required action may be wrongly interpreted as a violation leading to unfair penalties. An early identification of the absence of information can mitigate this risk by way of motivating discussions among participants about the required action and avoid unpleasant and inequitable situations.

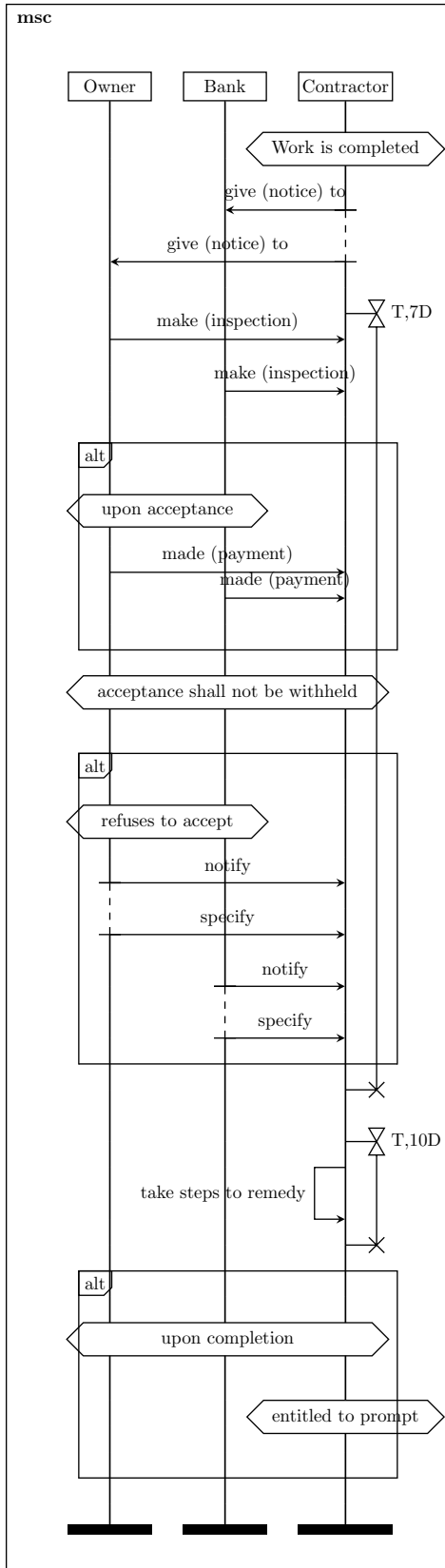


Figure 1: MSC depicting an example governance process

3.3 Render Layer

Once the mentions of the above types of entities are extracted and mapped to the governance process as described above, we use a simple visual (yet mathematically rigorous) notation, - MSC to depict the process. Fig. 1 depicts the example governance process depicted as an MSC.

The MSC notation is an international standard [Mauw, 1997] and is similar to the sequence diagram notation in UML. We chose MSC because, unlike other notations for representing business processes (for e.g., BPMN and BPEL), MSC has a clearer and simpler rigorous semantics, making formal analysis and verification of process models much easier. Interactions between actors are mapped to *messages* (directed arrows) in the MSC, where the initiator becomes the sender of the message and the recipient becomes the receiver of the message. We label each message with the associated action in the governance process. We map multiple messages to a *co-region* in the MSC on the sender's timeline, when it is not clear in which order the actor sends these messages. We map each trigger to a *condition* in the MSC formalism, which is labeled with text that indicates a state of the actor(s) and which is depicted as a hexagon containing the text of the trigger. *Timer* facility in the MSC formalism is used to denote whenever something is required to happen within a stipulated time (*deadlines*). MSC allows the ALT-box notation to specify alternate paths in the interactions among actors.

The MSC notation visually shows a *timeline* for each actor, in which the events (i.e., messages) that the actor participates in are depicted chronologically. Thus, the interactions depicted on a particular actor's timeline need to be temporally ordered. While text order (i.e., the order in which interactions are mentioned in the text), is a good initial approximation, in reality, many constructs in English can be used to alter the position of some interaction on the actor's timeline. We have designed an ILP-based event ordering algorithm, which maps the event ordering problem to an optimization problem, and the optimal solution is used to depict these interactions on the actors' timelines.

The algorithm to generate MSCs deals with a number of issues such as actor *co-reference* (an actor may be mentioned in many different ways, including pronouns), identifying the sender (i.e., initiator) and receiver(s) (i.e., recipients) for each action, and temporal ordering of messages. When the sender (or receiver) cannot be inferred, our algorithm uses *environment* as a generic actor for this purpose. Actions can also be co-referenced (e.g., upon acceptance thereof), which is the well-known NLP problem of *event co-reference*. For details of MSC extraction algorithm, see [Palshikar *et al.*, 2019a; Palshikar *et al.*, 2019b].

3.4 Related Work

Disambiguation of complex documents such as regulations and contracts has been an evolving research area. Researchers have approached this problem from various angles such as extracting obligations and rights [Breux *et al.*, 2006], identifying and classifying ambiguities [Massey *et al.*, 2014], integrating information from multiple sources to make sense of ambiguous texts [Ghaisas *et al.*, 2018] and formalizing contracts [Lomuscio *et al.*, 2012]

Information extraction (IE) from documents has been employed extensively for uncovering important details present in complex documents thereby disambiguating them [Palshikar, 2012]. Most of the text analytics work on contracts [Lippi *et al.*, 2017; Indukuri and Krishna, 2010; Gao and Singh, 2014; Curtotti and McCreath, 2010; Gao *et al.*, 2012; Hachey and Grover, 2004] is focused on extracting either entire sentences or clauses, rather than extracting specific contract elements. We found very few references on extraction of specific contract elements.

Gao and Singh [2014] proposed a topic modeling based technique to extract business events and their temporal constraints from contract text. Since our purpose is to disambiguate complex text, we extract elements at a much finer granularity, reason over the extracted elements and render them as a user-friendly visual depiction. Kalia *et al.* [2013] proposed an approach for extracting commitments from email and chat conversations. Their approach deals with ad hoc communications unlike our approach where the focus is on legal contracts and ours is a complex problem given that legal texts are written in natural language in its full complexity. Chalkidis and Androusoopoulos [2017] employed deep learning methods to extract various contract elements. They extract 11 elements namely Title, Party, Start, Effective, Termination, Period, Value, Gov. law, Jurisdiction, Legislative reference and headings. However, they neither reason nor render it the way we do. Biagioli *et al.* [2005] extracted provision types (Definitions, Obligations etc.) and their arguments (for instance, for obligations, the arguments are Addressee, Action, Third party) from law documents. This work of Biagioli *et al.* [2005] is close to ours as they also extract obligations, actions and actors. However, they do not extract conditions (Triggers in our case) and further they do not aim to render them in a user-friendly visual depiction.

4 Conclusion

We have been able to employ the R3 architecture to demonstrate that it is possible to recognize the crucial information elements in a contracts document, reason over them to determine their interrelationships with a high accuracy and render the result of this reasoning in a user-friendly visual form using MSCs. Importantly, the MSC representation also lets us identify ambiguities and inconsistencies in the text thereby allowing for informed discussions around contractual clauses that likely to remain unresolved in meaning and lead to unintentional violations and painful punitive actions.

Given the challenges and the crucial digital disruption happening in the financial domain, we took a contracts document related to financing construction for our experiments. Since our approach does not rely on any domain specific ontologies, it is domain independent and therefore is generalizable across domains.

In disambiguating contracts and presenting them as MSCs, we take a significant step towards making it easier for participants to fulfill their respective obligations. We will next validate the acceptability of the rendering with practitioners to gain their insights on determining the future direction of this work. Further, we would apply our extraction technique

on larger datasets from other domains to strengthen our generalizability claim.

References

- [Scardovi, 2017] Scardovi, C., 2017. Digital transformation in financial services. Springer.
- [Veerkar, 2018] Anand Veerkar, 2018. Why the Banking Industry Needs Enterprise Contract Management Software, <https://www.icertis.com/blog/banking-contract-management/>, Last accessed on 9-04-2019
- [Massey *et al.*, 2014] A. K. Massey, R.L. Rutledge, A.I. Antón, P.P. Swire, Identifying and classifying ambiguity for regulatory requirements, *In Proc. Of 22nd IEEE international Requirements Engineering Conference (RE)*, 2014, pp. 83-92
- [Reuters, 2017] From gadget news NDTV at: <https://gadgets.ndtv.com/mobiles/news/qualcomm-sues-apple-for-breach-of-contract-says-it-shared-information-with-intel-1770592>, Last accessed on 09-04-2019
- [Mauw, 1997] S. Mauw, 1997. ITU-TS Recommendation Z. 120: Message Sequence Chart (MSC).
- [Palshikar *et al.*, 2019a] G.K. Palshikar, N. Ramrakhiani, S. Patil, S. Pawar, S. Hingmire, V. Varma, P.Bhattacharyya, Extraction of Message Sequence Charts from Software Use-Case Descriptions, *The 2019 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT 2019 Industry Track)*, 2019, Minneapolis, USA.
- [Palshikar *et al.*, 2019b] G.K. Palshikar, S. Pawar, S. Patil, S. Hingmire, N. Ramrakhiani, H. Bedi, P. Bhattacharyya, V. Varma, Extracting Message Sequence Charts from History Narratives, *Workshop on Narrative Understanding co-located with NAACL-HLT 2019*.
- [Patil *et al.*, 2018] S. Patil, S. Pawar, S. Hingmire, G.K. Palshikar, V. Varma, P. Bhattacharyya, Identification of Alias Links among Participants in Narratives, *Association of Computational Linguistics (ACL) 2018*.
- [Breux *et al.*, 2006] T. D. Breux, M. W. Vail, and A. I. Anton, A.I., Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. *In Requirements Engineering, 14th IEEE International Conference* (pp. 49-58). IEEE.
- [Ghaisas *et al.*, 2018] S. Ghaisas, A. Sainani, A. P.R. Anish, P.R., 2018, Resolving ambiguities in regulations: towards achieving the Kohlbergian stage of principled morality, *In 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pp. 57-60. IEEE
- [Palshikar, 2012] G. K. Palshikar, Techniques for named entity recognition: a survey, in S. Bruggemann, C. D'Amato (Ed.s), *Collaboration and the Semantic Web: Social Networks, Knowledge Networks and Knowledge Resources*, IGI Global, 2012, pp. 191-217.

- [Lippi *et al.*, 2017] M. Lippi, P. Palka, G. Contissa, F. Lagoia, H. W. Micklitz, Y. Panagis, G. Sartor, P. Torroni, Automated Detection of Unfair Clauses in Online Consumer Contracts, *Legal Knowledge and Information Systems*, 2017, p.145.
- [Indukuri and Krishna, 2010] K. V. Indukuri and P. Radha Krishna, Mining e-contract documents to classify clauses, in *Proc. of the 3rd Annual ACM Bangalore Conference (Compute)*. Bangalore, India: ACM, 2010, pp. 1–5.
- [Gao and Singh, 2014] X. Gao, M. P. Singh, Extracting normative relationships from business contracts, *In Proc. of the international conference on Autonomous agents and multi-agent systems*, 2014, pp. 101-108
- [Curtotti and McCreath, 2010] M. Curtotti, E. McCreath, Corpus based classification of text in Australian contracts, *In Proc. of the Australasian Language Technology Association Workshop*, pp. 18–26, Melbourne, Australia, 2010
- [Gao *et al.*, 2012] X.Gao, M.P.Singh, P.Mehra, “Mining business contracts for service exceptions”, *IEEE Transactions on Services Computing*, 5, pp. 333–344, 2012
- [Hachey and Grover, 2004] B. Hachey, C. Grover, Sentence classification experiments for legal text summarisation, *In Proc. 17th Annual Conference on Legal Knowledge and Information Systems (Jurix-2004)*, pp. 29-38.
- [Gao and Singh, 2014] X. Gao, M.P. Singh, Mining contracts for business events and temporal constraints in service engagements, *IEEE Transactions on Services Computing*, 2014, 7(3), pp.427-439
- [Chalkidis and Androutsopoulos, 2017] I. Chalkidis, I. Androutsopoulos, A Deep Learning Approach to Contract Element Extraction, *In JURIX*, 2017, pp. 155-164
- [Biagioli *et al.*, 2005] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, C. Soria, “Automatic Semantics Extraction in Law Documents”, *In Proc. of the 10th International Conference on Artificial Intelligence and Law*. Bologna, Italy, 2005, pp. 133–140
- [Strotgen and Gertz, 2010] J. Strotgen, M. Gertz, HeidelbergTime: high quality rule-based extraction and normalization of temporal expressions, *Proc. Fifth Int. Workshop on Semantic Evaluation, ACL 2010*, pp. 321-324, 2010
- [Kalia *et al.*, 2013] A. Kalia, H. Motahari Nezhad, C. Bartolini, M. Singh. Monitoring commitments in people-driven service engagements. *Proc. SCC*, pp. 160–167 2013
- [Lomuscio *et al.*, 2012] A. Lomuscio, H. Qu, M. Solanki. Towards verifying contract regulated service composition. *JAAMAS*, 24(3):345–373, 2012

Rationale classification for educational trading platforms

Annie T.T. Ying¹, Pablo A. Duboue²

¹Cisco Vancouver AI Lab*

²Textualization Software Ltd.
Vancouver, Canada

Abstract

Stock market trading simulation platforms have become popular finance education tools in recent years. To encourage students to think through a trade order, many of such platforms provide a field called “rationale” in the trade order user interface. In this paper, we first present a novel problem called “thoughtful rationale classification” based on two studies: (1) an observational study on the factors affecting a finance professional assessment of a student’s trading sophistication and (2) a qualitative study on 2,622 rationales. The two studies together reveal that when a student provides thoughtful rationales, defined as rationales that document external research, specific strategies, or any technical analysis performed, the student is likely to be assessed higher in terms of the trading sophistication. We show that labelling rationales as thoughtful or not is a well defined task and automate it using CNNs. We also compare baseline implementations using simple features and support vector machines over selected keywords.

1 Introduction

Stock market trading simulation platforms have become popular finance education tools in recent years. These platforms provide trading capabilities such as buying/selling (as well as shorting and covering short) on a variety of securities (equities, bond, options, futures), providing a realistic hands-on experience for a student learning to trade.

To encourage students to think through a trade order, many of such platforms provide a field called *rationale* in the trade order user interface. To go one step further, some of such platforms allow a professor to specify the *minimum of rationales* a student has to provide in order to get a participation score as part of their grade. However, simply requiring the minimum number of rationales does not reward a student who provides a thoughtful rationale such as¹

XX is expected to show strong performance following Trump’s increased defence spending \$YY billion

from a student who provides rationales such as “good company,” “good stock,” or “the stock is rising.” A *thoughtful rationale* documents external research, specific strategies, or any technical analysis performed, as opposed to a rationale that lacks any type of thought or analysis.

In this paper, we first present a novel problem called *thoughtful rationale classification* based on two studies: (1) an observational study on the factors that affect a finance professional’s assessment on a student’s trading sophistication and (2) a qualitative study on 2,622 rationales from a trading simulation platform called EquitySim.² The two studies together reveal that when a student provides *thoughtful rationales*, a student is likely to be assessed higher in terms of the trading sophistication. On the other hand, a student who provides rationales that lack any type of thought or analysis are likely to be assessed as lower in trading sophistication. More importantly, there is evidence that introspection and retrospection lead to better learning [Koh *et al.*, 2018]. From this perspective, encouraging a student to write less trivial rationales will help a student learn irrespective of whether there is a correlation between better rationales and better trading performance.

The proposed thoughtful rationale classification is useful in a number of use cases:

- Trading sophistication assessment — Thoughtful rationales were consistently used as a trading sophistication marker from the observational study.
- Teaching good behaviour — Using this classifier can provide immediate feedback to a student to encourage more thought into a trade order.
- Engagement assessment — Counting the number of thoughtful rationales is a much better engagement metric than simply the number of rationales provided.

The contribution of the paper are as follows:

- We show that labelling rationales as thoughtful or not is a well-defined task.

*Work done at EquitySim, Inc. and prior to joining Cisco.

¹The examples are paraphrases of real rationales.

²<https://equitysim.com/>

- We automate this task using CNNs and compare it against several baseline systems.

This paper is organized as follows: in the next section, we discuss the observational study that motivated the creation of the task. In Section 3, we discuss the annotation and precise definition of the task. Section 4 discuss the different approaches we employed to solve the task, together with the evaluation results. We briefly touch on related work to our task in Section 5. Discussion of future work concludes the paper.

2 Motivating Thoughtful Rationales: Observational Study

In this section, we present an observational study which provided the motivation on the importance of classifying thoughtful rationales. The observational study involved an industry expert reviewing the overall trading sophistication of 11 randomly chosen subjects on a trading simulation platform, based on the past trading actions. Some of these trading actions include the type of security types (equities, bond, option, or future), the amount of the trade, and the rationale text provided along with a trade order.

The evaluator reviewed the past trading actions using the think-aloud protocol [Lewis and Rieman, 1993], i.e., verbalizing the thought process so that the transcript and systematic notes could be taken. This study is exploratory and qualitative in nature; thus, the evaluator was not given a specific set of instruction when they walked through the trading activities, other than “who are good” and “why.”

To analyze the factors affecting the evaluation of past trading actions, we used a grounded theory approach [Creswell, 2008], assigning codes to the part of the transcript wherever the evaluator mentioned factors indicative of trading sophistication. The following are the factors.

Portfolio’s risk and return (8 subjects)

Portfolio management, from the perspective of modern portfolio theory [Grinold and Kahn, 2000], is about the optimal management of risk and returns. A large positive return is obviously good but luck can be at play. Especially for assessing students’ portfolios, return should not be the only factor being considered in evaluating trading sophistication. The evaluator in 8 out of 11 subjects explicitly mentioned risk and return, as well as their trade-offs. Some examples the evaluator explicitly commented on the subjects’ portfolios are “good looking chart, steady, good return” (Subject 2); “risky, 2.42 beta which is quite high, but he delivers the return” (Subject 3), and “one big dip in one shot, big return” (Subject 4).

Portfolio diversification (7 subjects)

Diversification is one of the most fundamental strategies in portfolio management for mitigating risk. The evaluator in 7 out of 11 portfolios verbalized aspects of the portfolio related to diversification. For example, the evaluator commented that Subject 3’s portfolio is “not good, 81% in one company which is not good, and lots of companies but tiny proportions,” whereas for Subject 2, “industry pretty wide but still may not diversify properly but it’s OK.”

Rationale text (7 subjects)

The evaluator mentioned trade rationales in 7 out of 11 subjects explicitly. For example, on Subject 3, the evaluator lamented “Bad rationales like bullish, good day, rising” whereas on Subject 9, the evaluator commented “better rationales mentioning earning reports.”

This is the context that motivates us to look into rationales and see whether we can construct a well-defined task for natural language processing, as described in Section 3.

Complex instruments (6 subjects)

Using complex investment instruments such as options and futures, in the student assessment context, is a sign of sophistication. For example, the evaluator commented on Subject 1 of having “futures: the riskiest thing, but a sign of sophistication” and Subject 2 on having “a bit of options, good.”

Trading strategy (2 subjects)

The evaluator in two cases inferred the trading strategy. e.g., “sell when they are down to manage risk, have ranges in mind” (Subject 2) and “large bounds, a waiter, a tolerant trader” (Subject 9).

3 Thoughtful Rationales Definition and Labeling

To elucidate what would be useful to extract from trade rationale text, we conducted a manual analysis on a set of 2,622 rationales within the past trading meta-data from a random sample of 35 users from the EquitySim trading simulation platform. The manual analysis is based on an inductive, qualitative approach.

We found that the thoughtfulness of a rationale is fruitful to focus on for a natural language processing task: four levels of rationale thoughtfulness emerged from the manual analysis of the 2,622 rationales, ranging from a rationale containing little or no thought, to one containing an extensive amount of research or analysis. We chose not to focus on whether a rationale was factually correct—for example, if the rationale mentioned an earnings number, whether the number was actually correct—as this task is a significantly more challenging task for human to annotate, and also significantly more challenging to automate.

The rest of this section provides a precise definition of the task (Section 3.1) and the annotation guide (Section 3.2 and Table 1) constructed for the initial annotation of the 2,622 rationales (866 after de-duplication).³ These 866 deduplicated rationales are our core evaluation data. Note we do not split the evaluation by users, mixing rationales from different users in the test and training sets. A more rigorous evaluation splitting at the user level is possible, but it was not investigated in the present work.

3.1 Definition

The four levels of rationale thoughtfulness emerged from the manual analysis of 2,622 rationales are as follows:

³As the annotation was done in batches over the different students, de-duplication was not considered an issue until later in the process.

Table 1: Types of content among the four levels of thoughtful rationale and examples

Type of content	L1	L2	L3	L4	Examples
Too general	244				“Good trade”, “Investment”, “To make money”
Non-sensical text	103				“oops”, “No laughter... pen drop!!!”
Non-sensical chars	68				“asdfasdfasdf”
Bug	3				“The platform sold my position accidentally. Rebuying”
General price movements		1446			“stocks are rising”, “bullish”, “short term correction”
General company fundamentals		63			“The company is taking an exciting new direction!” “Expected news”
General economical and & political fundamentals		9			“Economy looking strong. Market is on a rally.” “trump stock”
General comment on portfolio strategy (e.g., allocation)		13			“Diversify”, “Pharmaceuticals”, “Government Bond”, “Oil prices rising hopefully”
Uncertain, mistake		29			“this was a gamble”, “accidentally hit the trade button”
Technicals			81	18	Level 3: “Day range being tested”, “XX Finance has been performing well.” Level 4: mentioning mathematical calculations e.g., moving averages, Doji, Bollinger bands, inflection point
Fundamentals - company (e.g., earnings, news, structural events such as M&A, IPO)			127	188	Level 3: “I believe in the stability of this company due to its long history” “Growth with attractive valuation”, “Good earnings upside”, “Reveal New Product” Level 4: “Monopoly as in-flight internet service provider, solid fundamentals. Betting to go up after earnings”, “XX failed merger depressing price”, “XX earning reports: bottom line YY% increase; investor expectations were too high”, “XX Accused of Gouging Customers on Prescription Drugs”
Fundamentals - economical & political (e.g., interest rates, seasonal events, elections)			39	29	Level 3: “possible rate increase next week” “Projecting market to fall after election” “cyber monday sell off” Level 4: “XX vote for rate hikes will hurt euro value” “higher XX prices in future months especially with the possible uk EU exit.”
Portfolio strategy (e.g., asset allocation, diversification, portfolio actions, limit orders, hedging, target price)			30	37	Level 3: “XX assets because they provide very stable cash flows over the long term”, “Diversification across markets”, “covering short. taking profits” Level 4: “XX has a virtual monopoly. Stock is undervalued. Target price=XX mid April.”
External advice / personal experience			13	4	Level 3: “Buffett growth outlook” Level 4: “XX said: ‘hold’ to ‘buy’ + long term uptrend”
Total	418	1560	290	276	

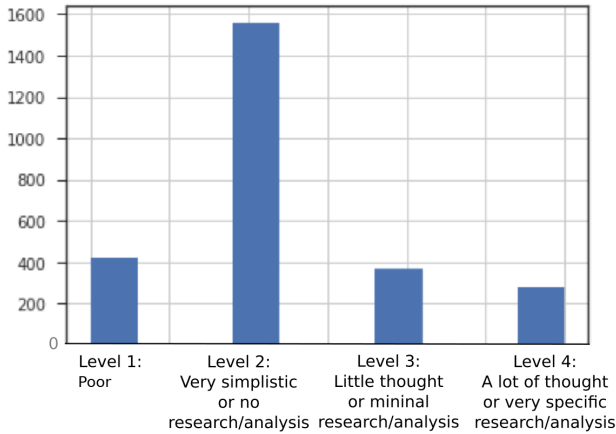


Figure 1: Distribution of rationales by levels of thoughtfulness

1. A rationale contains little or no thought.
2. A rationale contains research or analysis that is simplistic or too general.
3. A rationale contains specific research or analysis.
4. A rationale contains an extensive amount of research or analysis.

Figure 1 shows the distribution of the four levels of thoughtfulness.

3.2 Annotation Guide

Table 1 presents the results emerged from the manual analysis for each of the four thoughtfulness levels. The table and the rest of Section 3.2 serve as the annotation guide for this task. For each thoughtfulness level, Table 1 presents the type of content would be considered at what level (column “Type of Content”) and the corresponding number of rationales for each of the four levels (columns “L1”, “L2”, “L3”, and “L4”), as well as example rationales (column “Examples”).

A level 1 rationale contains no or little thought.

Most of the rationales in level 1 are either too general or even non-sensical. There were a small portion of rationales that spoke about bugs users experience in the system.

A level 2 rationale contains simplistic or general research or analysis.

Though these rationales are better than the level 1’s, the content still does not demonstrate any specific research or analysis. The large majority of these rationales (1,449) are simplistic reasons on the price movements and predictions, such as “stocks are rising.” Level 2 rationales can also describe either company or economical/political fundamentals, though without much specific content, such as “The company is taking an exciting new direction!”

A level 3 or 4 rationale contains specific research or analysis, with level 4 having a significant amount.

For level 3, there were 81 rationales with content focusing on the technicals; for level 4, there were 188 rationales including specific mathematical indicators mentioned in the rationale

text such as moving average [Bauer and Dahlquist, 1998], Doji [Bauer and Dahlquist, 1998] and Bollinger bands [Bauer and Dahlquist, 1998]. Additionally, there were 127 level 3 rationales and 29 level 4 rationales on the fundamentals of a company, such as structural aspects (e.g., merge and acquisitions), earnings or results, or news on a company. There were 39 level 3 rationales and 29 level 4 on economical fundamentals such as interest rates and other economical events.

We found that 85 rationales contain more than one type of contents: e.g., “Bottom, possibility for some M&A.”

3.3 Inter rater agreement

To gauge the complexity of the task and the quality of the instructions, a second person, not familiar with the work, nor with trading concepts annotated 25 rationales in the two class category task (thoughtful vs. non-thoughtful, conflating levels 1-2 and 3-4 above).

Out of a chance agreement of 12.6, both annotators agreed in 19 times, for a Kappa statistic [Carletta, 1996] of 0.516 (a “moderate” agreement). The differences hinged in the understanding of technical terms, for example “covering” expressed an important trading concept that was lost to the second annotator.

This moderate agreement is encouraging and highlights either the need of expert annotators or, if we were to move to the use of crowd workers, the need of machine learning that can profit from noisy labels.

4 System

We experimented with five systems: two sanity baselines (Section 4.1), one SVM baseline (Section 4.2), and two deep learning models (Sections 4.3 and 4.4). The training data for the deep learning models consists of a set of 866 rationales (after de-duplication from 2,622) annotated as described in Section 3. The four thoughtfulness levels are conflated into two labels, *thoughtful* being level 3 or 4, and *not thoughtful* being level 1 or 2. The final dataset we are using has 403 rationales labeled as not thoughtful and 486 as thoughtful.

4.1 Sanity baselines

We approached the problem using existing tools. We started with two sanity baselines. The first sanity baseline uses two features of the rationale: its length and whether it contains a digit. If the length in characters was greater than a fixed threshold (30 characters) or if it contains a digit, then we consider the rationale to be thoughtful. This baseline achieves a precision of 0.812, a recall of 0.725, and a F-measure of 0.766 (Table 2). The second sanity baseline requires both signals to be present (length and digit). This baseline achieves a significantly higher precision of 0.988 but only a recall of 0.182, resulting in a relatively low F-measure of 0.308 overall. As neither of these baselines is trained, the results are over all the 866 rationales.

4.2 SVM

Next, we wonder whether the length threshold could be learned and whether the number of tokens besides the number of character might make for an informative feature. Moreover, using a broad lexicon is traditionally associated with

Table 2: Evaluation results

System	Prec	Rec	F1
Baseline	81.2	72.4	76.5
Strict Baseline	98.9	18.2	30.7
SVM	66.3	81.5	72.7
CNN	82.1	87.1	84.4

textual sophistication in language learning, so we added a third feature indicating the average IDF score for the top 3 highest IDF words in the rationale. As an IDF source, we used 21,000 articles from Thompson Reuters, for a total vocabulary of 133,000 word tokens. Training a Support Vector classifier using a Gaussian kernel on these three features (length in characters, length in tokens, average of the top 3 IDF scores) produced better recall than the baseline system at the expense of lower precision, for a diminished F-measure (third row in the table, evaluated using 10-fold cross-validation).

4.3 CNNs

We then trained a deep learning model for this text classification task using 10-fold cross-validation. The text classification model is trained using a convolutional neural network (CNN) [Goodfellow *et al.*, 2016] with residual connections [Honnibal, 2016] from the spaCy library [Honnibal and Montani, 2017]. The model assigns pre-trained position-sensitive vectors provided by the spaCy library to each word in a rationale. The document tensor (on a rationale) is produced by concatenating max and mean pooling, and a multi-layer perceptron is used to predict an output vector, before a logistic activation is applied to each element. The value of each output neuron is the probability of the class (thoughtful or not). The neural network architecture is similar to the hierarchical attention network [Yang *et al.*, 2016] which has two levels of attention mechanisms applied at the word- and sentence-level. The difference with Yang *et al.*'s model is in the word embedding strategy (which uses sub-word features and Bloom embeddings [Serrà and Karatzoglou, 2017]) and that CNNs are used instead of BiLSTMs. Because the rationales are all single sentence, the sentence-level attention does not play a role in our case. But the word-level attention is definitely important.

4.4 Transfer learning

We decided to profit from 13,000 unannotated rationales and the strong baseline to experiment with multi-task learning [Caruana, 1997] and transfer learning using universal language models [Howard and Ruder, 2018]. For multi-task learning, we trained a LSTM and two dense layers for three tasks (Figure 2): the thoughtful rationale prediction baseline, predicting the type of operation (buy vs. sell and short) and the type of instrument (stock vs. options and bonds). The hope was the extra tasks will help make the network less tuned to just counting characters and detecting digits, as the network would be if it were trained only the baseline task.

For input, we used spaCy tokenization and word2vec embeddings [Mikolov *et al.*, 2013] over the 21,000 articles from

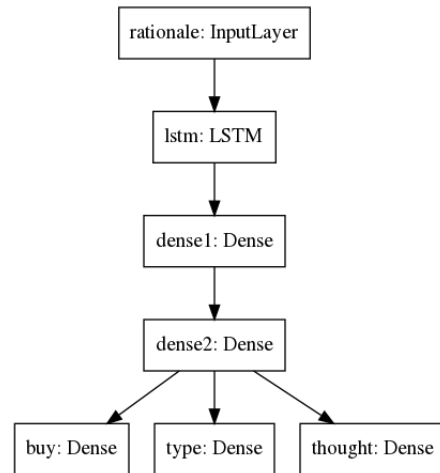


Figure 2: Multi-task network architecture

Thompson Reuters described before. We used an embedding size of 50 dimensions with continuous-bag-of-words model. The total vocabulary size was 80,263 embeddings. When transforming the rationales into embeddings, due to the more colloquial, informal aspect of the input, there are many misspellings. Of the 321,538 different word types present in the input rationales, 18,058 are missing.

The parameters used are as follows:

- Input sequences were truncated to 40 tokens.
- The LSTM used a 50-unit memory.
- First dense layer used a 50 neurons, with ReLU activation.
- The second dense layer used 25 neurons, with ReLU activation.
- The output layers used a single neuron, with sigmoid activation.
- The network was trained for 12 epochs using a batch size of 256.

We experimented with different layer sizes and drop-out but these were the optimal parameters we obtained for this task.

This network was then transferred using only the thoughtful rationale task over the 866 annotated rationales. The evaluation was done as the average over ten bootstraps using 20% held-out for evaluation. We used a slanted triangular learning rate with a base learning rate of 0.004 and a maximum learning rate of 0.01, on top of the Adam optimizer [Howard and Ruder, 2018], plus step-wise thawing of the layers. We used 5 epochs with a batch size of 64. The best accuracy we achieved was 83.3% (average over ten bootstraps), which is below what we obtained using spaCy's CNNs. And that is using an optimal architecture that is risking over-fitting. We continue to investigate other approaches.

4.5 Some qualitative validation

Finally, interviews with professors provided some initial positive qualitative evidence: For example, a professor described

a student with the most number of thoughtful rationales in his class as someone “you can always find him in our building reading the Wall Street Journal.”

5 Related Work

The most common application of natural language processing over financial texts focus on sentiment analysis of on-line discussion around particular stocks [Bollen *et al.*, 2011; Pagolu *et al.*, 2016]. Text classification has also been applied to the finance domain to solve a variety of problems like predicting a firm’s credit risk [Byanjankar *et al.*, 2015] and compliance [Fisher *et al.*, 2016], and evaluating loan application [Netzer *et al.*, 2018]. Here, we have taken an approach more closely aligned with natural language processing applications for language education [McNamara *et al.*, 2013; Kyle and Crossley, 2015; Gao *et al.*, 2018]. In that setting, trading rationale sophistication can be considered similarly to existing techniques to automatically evaluate the quality of an essay or a dialogue turn.

The type and quality of the language exhibited by a student has been gaining attention in the field of Educational Data Mining (EDM). Recent work by Crossley *et al.* [Crossley *et al.*, 2018] has tied linguistic features in the language used to talk with a pedagogical agent with their Math Identity. The concept of Math Identity [Nosek *et al.*, 2002] expresses how much of a “math person” the student identifies themselves. We have not taken such a deep analysis route ourselves with the thoughtful rationale identification but we find it an exciting direction. It is possible the type of features used by Crossley and colleagues can be useful for our task, too. It is also enriching to consider a Trading Identity similar to the Math Identity.

Our setting involves a human (the student) making complex decisions (trading) and explaining their actions through rationales. A mirror setting involves a computer system (an agent) interacting with a complex world and explaining its actions [Johnson, 1994; Lacave and Díez, 2002; Stumpf *et al.*, 2009; Lei *et al.*, 2016]. While the connection between the two topics is tenuous and we have not explored in our work, it might be possible to use our available data to automatically produce rationales. They can be used to show the student what a quality rationale for their trade might be and to make them double check their assumptions. Using training data for a generation system poses challenges, though, as poor quality training text will make for generator that produces poor quality output text [Reiter and Sripada, 2002]. In this setting, our work can be seen as pre-filtering quality text that can be then used to build an automatic explanatory system.

6 Conclusion

We have presented here the concept of *thoughtful rationales*, defined as a rationale that documents external research, specific strategies, or any technical analysis performed and showed that our analysis and discussion with professors indicate it is an indicator of trading sophistication. We then proceeded to automate the identification of thoughtful rationales through classification systems and experiments.

Some future improvements on our rationale classification system include features from the trade order, such as the security, trade amount, security type, and time of the order. We also want to explore new transfer learning models such as BERT [Devlin *et al.*, 2018].

A more challenging task would be to determine whether a rationale text displays an additional level of inference. For example, buying a certain stock based on an earnings report is simply reacting to the market where the news is already priced in. The highest level of a rationale would be one that involves an additional level of inference beyond simply reacting to the market, or a rationale that has a unique point of view differs from what the market expects.

Another useful task is to classify whether a rationale contains content on company fundamental, economic fundamental, or a technical piece of analysis. This is ongoing work on evaluating trading sophistication, where rationale thoughtfulness is a strong indicator.

Finally, we might want to expand the system to explore issues of Trading Identity or even a full-fledged trading rationale generation.

Acknowledgements

We would like to thank Justin Ling for support and sharing his finance insights in the observational study. We would also like to thank the reviewers and Denys Gajdamaschko for their useful questions and feedback.

References

- [Bauer and Dahlquist, 1998] Richard J Bauer and Julie R Dahlquist. *Technical Markets Indicators: Analysis & Performance*, volume 64. John Wiley & Sons, 1998.
- [Bollen *et al.*, 2011] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.
- [Byanjankar *et al.*, 2015] Ajay Byanjankar, Markku Heikkilä, and Jozsef Mezei. Predicting credit risk in peer-to-peer lending: A neural network approach. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 719–725. IEEE, 2015.
- [Carletta, 1996] Jean Carletta. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.*, 22(2):249–254, 1996.
- [Caruana, 1997] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.
- [Creswell, 2008] John W. Creswell. *Research design: Qualitative, Quantitative, and Mixed Methods Approaches*, chapter Chapter 9: Qualitative Procedures. Thousand Oaks: Sage Publications, 2008.
- [Crossley *et al.*, 2018] Scott Crossley, Jaelyn Ocumpaugh, Matthew Labrum, Franklin Bradfield, Mihai Dascalu, and Ryan S Baker. Modeling math identity and math success through sentiment analysis and linguistic features. *International Educational Data Mining Society*, 2018.

- [Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [Fisher *et al.*, 2016] Ingrid E Fisher, Margaret R Garnsey, and Mark E Hughes. Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research. *Intelligent Systems in Accounting, Finance and Management*, 23(3):157–214, 2016.
- [Gao *et al.*, 2018] Yanjun Gao, Patricia M Davies, and Rebecca J Passonneau. Automated content analysis: A case study of computer science student summaries. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 264–272, 2018.
- [Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, chapter Chapter 9: Convolutional Networks. MIT Press, 2016.
- [Grinold and Kahn, 2000] Richard C Grinold and Ronald N Kahn. Active portfolio management. 2000.
- [Honnibal and Montani, 2017] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing, 2017.
- [Honnibal, 2016] Matthew Honnibal. Embed, encode, attend, predict: The new deep learning formula for state-of-the-art nlp models, 2016.
- [Howard and Ruder, 2018] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, 2018.
- [Johnson, 1994] W Lewis Johnson. Agents that learn to explain themselves. In *AAAI*, pages 1257–1263, 1994.
- [Koh *et al.*, 2018] Aloysius Wei Lun Koh, Sze Chi Lee, and Stephen Wee Hun Lim. The learning benefits of teaching: A retrieval practice hypothesis. *Applied Cognitive Psychology*, 32(3):401–410, 2018.
- [Kyle and Crossley, 2015] Kristopher Kyle and Scott A Crossley. Automatically assessing lexical sophistication: Indices, tools, findings, and application. *Tesol Quarterly*, 49(4):757–786, 2015.
- [Lacave and Díez, 2002] Carmen Lacave and Francisco J Díez. A review of explanation methods for bayesian networks. *The Knowledge Engineering Review*, 17(2):107–127, 2002.
- [Lei *et al.*, 2016] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, November 2016. Association for Computational Linguistics.
- [Lewis and Rieman, 1993] C. Lewis and J. Rieman. *Task-Centered User Interface Design: A Practical Introduction*, chapter Chapter 5: Testing The Design With Users. Self-published, 1993.
- [McNamara *et al.*, 2013] Danielle S McNamara, Scott A Crossley, and Rod Roscoe. Natural language processing in an intelligent writing strategy tutoring system. *Behavior research methods*, 45(2):499–515, 2013.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [Netzer *et al.*, 2018] Oded Netzer, Alain Lemaire, and Michal Herzenstein. When words sweat: Identifying signals for loan default in the text of loan applications. *Columbia Business School Research Paper*, (16-83), 2018.
- [Nosek *et al.*, 2002] Brian A Nosek, Mahzarin R Banaji, and Anthony G Greenwald. Math= male, me= female, therefore math \neq me. *Journal of personality and social psychology*, 83(1):44, 2002.
- [Pagolu *et al.*, 2016] Venkata Sasank Pagolu, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi. Sentiment analysis of twitter data for predicting stock market movements. In *2016 international conference on signal processing, communication, power and embedded system (SCOPEs)*, pages 1345–1350. IEEE, 2016.
- [Reiter and Sripada, 2002] Ehud Reiter and S. Sripada. Should corpora texts be gold standards for NLG? In *Proceedings of Second International Conference on Natural Language Generation INLG-2002*, pages 97–104, Arden House, NY, 2002.
- [Serrà and Karatzoglou, 2017] Joan Serrà and Alexandros Karatzoglou. Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 279–287. ACM, 2017.
- [Stumpf *et al.*, 2009] Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies*, 67(8):639–662, 2009.
- [Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

CoFiF: A Corpus of Financial Reports in French Language

Tobias Daudert^{*†} and Sina Ahmadi[†]

Insight Centre for Data Analytics
National University of Ireland, Galway
{tobias.daudert, sina.ahmadi}@insight-centre.org

Abstract

In an era when machine learning and artificial intelligence have huge momentum, the data demand to train and test models is steadily growing. We introduce CoFiF, the first corpus comprising company reports in the French language. It contains over 188 million tokens in 2655 reports, covering reference documents, annual, semestrial and trimestrial reports. Our main focus is on the 60 largest French companies listed in France’s main stock indices CAC40 and CAC Next 20. The corpus spans over 20 years, ranging from 1995 to 2018. To evaluate this novel collection of organizational writing, we use CoFiF to generate two character-level language models, a forward and a backward one, which we use to demonstrate the corpus potential on business, economics, and management research in the French language.

The corpus is accessible on Github ¹.

1 Introduction

Current research approaches progressively use machine learning and artificial intelligence to derive knowledge from large amounts of data. With natural language processing (NLP) being a crucial part in this progress, knowledge extraction from textual data becomes increasingly important and the underlying fuel, texts, are a sought source. While general corpora exist for many languages such as The British National Corpus [Leech, 1992] for British English, the Corpus of Spoken Professional American English [Barlow, 2000] for American English, or the *Corpus de Français Parlé Parisien des années 2000* (CFPP2000) [Branca-Rosoff *et al.*, 2000] for French, domain-specific corpora are still lacking in many cases. Since transfer learning, particularly language models such as ELMo [Peters *et al.*, 2018] or BERT [Devlin *et al.*, 2018], is currently driving NLP research, large unlabeled corpora play a progressively important role. Examples are the 1 billion word benchmark [Chelba *et al.*, 2013], Wiki-103

[Merity *et al.*, 2016], or CommonCrawl².

Considering the domain of business and economics, especially for English, corpora such as the Wall Street Journal (WSJ) Corpus [Paul and Baker, 1992], the 10-k Corpus [Kogan *et al.*, 2009] and the 8-k Corpus [Lee *et al.*, 2014] are popular examples. However, no corpus to date deals with French texts in the field of economics and finance. Such an absence hinders the progress in applying NLP approaches on the textual data related to the financial sector in francophone countries, particularly France, Canada, Belgium and Switzerland. Hence, we present CoFiF, a corpus aggregating French organizational writing into a source to be analysed in the area of business, economic and management. CoFiF contains documents published by companies which have been part of the *Cotation Assistée en Continu* (CAC) 40³ since 2002. CAC40 contains 40 of the 100 largest companies by market capitalization of the stock exchange in Paris. Furthermore, the CAC40 is France’s main stock index and is dominated by French companies, thus, it can be taken as a representation of French companies in general. In addition, we included companies listed at the CAC Next 20 in the corpus. These companies are the 20 largest ones which are listed following the ones in the CAC40, hence, altogether both indices list the 60 largest French companies. The collected document types provide a comprehensive and factual overview of a company’s shape. In addition, their language can also be consulted in linguistic terms. Previous analyses of company reports for English have shown their effect on the financial markets, for instance, Kogan *et al.* linked 10-k reports to market volatility and Lee *et al.* used 8-k reports to predict stock price movement in terms of [up, down, stay] [Kogan *et al.*, 2009; Lee *et al.*, 2014].

The rest of this paper is organized as follows: we first present previously created French corpora, both general and specific in the field of economics and finance. Following a description of CoFiF in section 3, we evaluate our corpus using a language model in section 4. The paper is concluded in section 5.

^{*}Contact Author

[†] Equal first authors

¹<https://github.com/CoFiF/Corpus>

²<http://commoncrawl.org/>

³<https://www.euronext.com/en/products/indices/FR0003500008-XP/Market-Information>

	CAC40			CAC Next 20			All		
	#Tokens	#Sentences	#Reports	#Tokens	#Sentences	#Reports	#Tokens	#Sentences	#Reports
Annual	20141096	550142	587	2778348	57762	133	22919444	607903	720
Semestral	3988379	78810	410	3302992	63233	254	7291371	142042	664
Trimestral	655991	14091	108	745049	15145	228	1401040	29235	336
Ref. docs.	123238519	3252462	736	33699932	1073180	199	156938451	4325641	935
Total	148023985	3895505	1841	40526321	1209320	814	188550306	5104821	2655

Table 1: Number (#) of tokens, sentences, and reports ordered stock indices and report types.

2 Related Work

There is a plethora of corpora available for the French language, both for general purposes [Abouda and Baude, 2005; Eshkol-Taravella *et al.*, 2010; Content *et al.*, 1990; Guillot *et al.*, 2008; Kunstmann and Stein, 2006] and for specific tasks in NLP. Vincent and Winterstein developed a French corpus for sentiment analysis [Vincent and Winterstein, 2013]. Grabaretal *et al.* targeted reports published in the scientific literature or used in medical education to create a French corpus with clinical cases [Grabar *et al.*, 2018]. Mariani *et al.* presented the NLP4NLP Corpus containing scientific articles published over a period of 50-year in the field of speech and natural language processing in various languages, including French [Mariani *et al.*, 2018]. Mondada *et al.* provided the International Ecological Corpus of French (CIEL) which promotes comparative analysis in the field of linguistic ecology of spoken French in francophone countries [Mondada and Pfänder, 2016]. The Sequoia corpus [Candito and Seddah, 2012] is a syntactically annotated French corpus containing phrases from the French Europarl [Koehn, 2005], l’Est Républicain regional newspaper articles, French Wikipedia, and documents from the European Medicines Agency. Similarly, Martineau *et al.* presented a morphosyntactically structured and annotated corpus (MCVF) to study morphosyntactic variations based on time and social distribution [Martineau, 2008]. Targeting contemporary French, Benzitoun *et al.* [Benzitoun *et al.*, 2016] assembled the ORFÉO which contains 4 million and 6 million words of spoken and written French, respectively.

Regarding corpora in economics and finance for other languages, Kloptchenko *et al.* [Kloptchenko *et al.*, 2004] were the pioneers in producing a corpus based on organizational English content for sentiment analysis for stock market prices prediction. A significant resource is the 10-K Corpus [Kogan *et al.*, 2009] which is composed of 54,379 annual reports in English from 10,492 different companies covering a time interval from 1996 up to 2006. This corpus has paved the way for further tasks in economics and financial text analysis. Similarly, Lee *et al.* created a corpus of 8-k reports which is subsequently used for stock price prediction [Lee *et al.*, 2014]. Recently, Händschke *et al.* [Händschke *et al.*, 2018] introduced the JOCo corpus which contains 5,000 reports (282M tokens) of corporate annual and social responsibility reports from UK, German and US companies for the period of 2000 to 2015.

Despite the need, there have been few efforts in creating French corpora in economics and finance. Verlinde *et al.* targeted Belgian and French newspapers published from 1986

to 1992 to purposely create a corpus containing 1 million words to analyze word combinations used in economical discourse [Verlinde, 1997]. Similarly, Foltête focused on newspaper articles published in France and created a corpus to analyze how economical discourse changes with respect to distributional semantics and transformational grammar [Foltête, 1999]. Focussing on the differences between specialized and non-specialized texts, Cabré presented a multilingual corpus in the field of economics containing 78k words [Cabré, 2007]. Gautier described the construction of a corpus based on the economic articles published in Le Monde and Les Echos for the study of the relation of neologism and economic crises [Gautier, 2012]. Gallego introduces a comparable corpus, COMENEGO, in French and Spanish in business, for translation purposes and a discursive analysis approach based on metadiscourse [Gallego-Hernández, 2013]. According to the literature, no French corpus of significant size is provided in finance and economics so far. To bridge this gap, we introduce the first French corpus dealing with company reports.

3 Corpus Description

Our selection criteria are based on the coherence of the published documents in the field of economics and finance. We can categorize such documents in four types: reference documents (*documents de référence*), which are published annually, usually in the months following the end of the calendar year, and contain information regarding the financial situation and perspectives of a company; annual results (*résultats annuels*) which summarizes a company’s business and activities throughout the previous year; semestrial results (*résultats semestriels*) and trimestrial results (*résultats trimestriels*) which are similar to the annual reports except that they are published every six months and three months, respectively. We included reference documents since some companies also consider this document type as the annual report. Moreover, we found that most companies publish their reference documents on a regular basis. This is not the case for other document types, particularly trimestrial results. All the collected document types report financial results and provide information to financial analysts, institutional investors and individual shareholders. In France, the *Autorité des marchés financiers*⁴ (Financial Markets Regulator), ensures standardization in reporting financial results by requiring companies to follow a certain template.

Regarding the targeted companies, we focus on the CAC40 and the CAC Next 20. Both indices combined contain 60

⁴<https://www.amf-france.org>

of the 100 largest companies by market capitalization of the stock exchange in Paris, thus we found these an appropriate choice for our corpus. Furthermore, we consider reports published between 1995 and 2019.

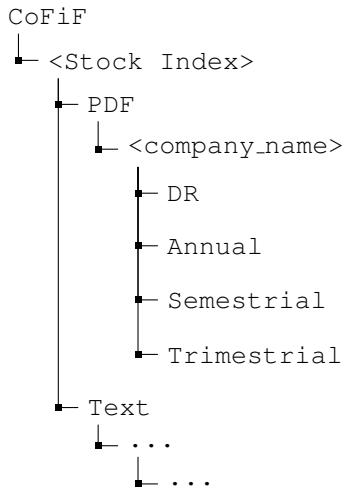


Figure 1: Structure of CoFiF

3.1 Data Retrieval and Corpus Structure

Having a strong and representative shortlist of companies in place, collecting documents was mainly done by consulting company’s website and downloading the reports since 1995. Although there are companies which provide such documents classified by year and type, in some cases the collection could be cumbersome due to lack of organization, website performance or non-continuous publishing (e.g. missing years). In other cases, a company might have changed the name during the last two decades, such as “Orange S.A.” (previously “France Télécom S.A.”), or is available under a new company name as outcome of a merge with another company such as the case of “Engie” (merged “Gaz de France” and “Suez”). Under these circumstances, we consulted the Web to find archives of previously published documents, particularly <https://www.bnains.org>, and included documents published under previous company names as well. Although the content of the financial document may be similar in most cases, such similarity is seen to a lesser extent in structure. Companies publish their financial information in various formats ranging from plain text and HTML to tabular and graphical representations. Given the availability of searchable Portable Document Format (PDF), we only included such documents in our corpus. The period of 20 years was chosen based on the availability of reports.

After downloading the reports, we extracted the texts with the command-line software `pdftotext`⁵ in UTF-8. We did not perform further preprocessing on the datasets as we believe that certain tasks, such as document structure extraction, may require different information which should not be affected by the preprocessing step. Nonetheless, we did not observe

⁵<http://www.xpdfreader.com/>

much noise in the collected text. To facilitate document processing, we provide meta-data in the structure of the corpus and the document names. Figure 1 illustrates the structure of CoFiF where documents are classified based on stock indices, CAC20 and CAC40, company name and document types. Further information regarding the publication date is provided in the file name. The structure and the file names in the PDF and Text directories are identical.

3.2 Corpus Analysis

In the following step, we conducted a corpus analysis with the help of the Natural Language Toolkit (NLTK) [Loper and Bird, 2002] for the segmentation and counting of tokens as well as sentences. Table 1 presents the results of the corpus analysis based on the document types and stock indices.

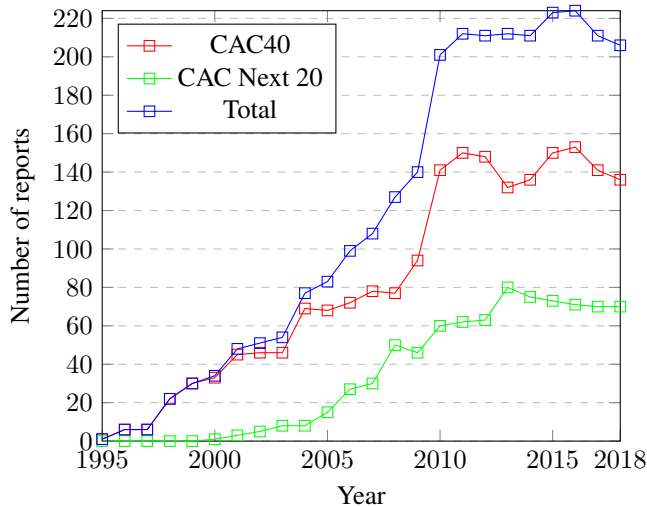


Figure 2: Distribution of reports per year

4 Experiments

To evaluate the corpus and show its potential for NLP tasks in French, particularly within the business and economic domain, we created two character-level language models (LM), trained using a forward recurrent neural network (RNN) and a backward RNN. To prepare the data for the LM, we removed repeated empty lines and breaklines, and aligned the content at the line beginning if necessary. Both language models are generated using a modified version of the NLP library `flair` [Akbik *et al.*, 2018; Akbik *et al.*, 2019]. To train both models we apply the following parameters: `hidden_size` 2048, `n_layers` 1, `sequence_lenght` 250, `mini_batch_size` 100, and `epochs` 3.

In addition to serving as a language model in its natural sense, it also provides word embeddings which can be used in downstream tasks such as text classification or sentiment analysis. The CoFiF word embeddings have shown their use in a sentence boundary detection task which achieved good performance obtaining an F1 score of 0.91 [Daudert and Ahmadi, 2019]. To evaluate both language models, we con-

Sentence	Perplexity
Perspectives d’avenir et principaux risques.	1.7892
Perspectives avenir et principaux risques.	2.9605
Le chiffre d’affaires de l’activité autocars augmente principalement suite à une amélioration du prix moyen, et ce malgré un recul des volumes de 3 %.	1.9745
Le chiffre d’affaires l’activité autocars augmente principalement suite de une amélioration du prix moyen, et ce malgré un recul dès volumes 3 %.	2.9471
Cette stratégie permettrait ainsi d’accroître les péages ferroviaires perçus par Groupe Eurotunnel pour l’utilisation de son infrastructure.	2.4411
Cette stratégie permettrait ainsi d’accroître les péages ferroviaires perçus par Groupe Eurotunnel que l’utilisation son infrastructure.	2.8991

Table 2: Six sample sentences and their perplexity scores retrieved by the character-level forward language model. The upper sentence of each pair is the original sentence, the lower sentence is the modified and wrong sentence.

bénéfice		perte		croissance		impôt		économie	
profit	0.715	dépréciation	0.666	progression	0.857	impôts	0.783	agriculture	0.672
versement	0.544	moins-value	0.599	décroissance	0.782	fonctionnelle	0.606	installation	0.609
solde	0.534	variation	0.571	hausse	0.731	imputation	0.592	énergie	0.603
résultat	0.512	insuffisance	0.515	amélioration	0.719	amortissement	0.535	problématique	0.593
dividende	0.512	diminution	0.505	dynamique	0.716	déduction	0.533	innovation	0.581

Table 3: Five sample word embeddings and their neighbours based on the cosine similarity.

ducted an experiment based on the sentence perplexity. First, we extracted 100 randomly chosen sentences from an annual report external to the corpus. Second, we duplicated these sentences and modified all 100 duplicates forcing grammatical and syntactical incoherences. In the following step, we calculated the sentence perplexity for each of the 200 test sentences. Lastly, we evaluated the correctness of the model’s predictions by using sentence perplexity scores. The model is correct when it returns a lower perplexity score for the initial sentence and a higher score for the modified sentence. The prediction is incorrect When the modified sentence receives the lower score. From the 100 test sentence pairs, our model detected all 100 original sentences correctly. Examples are presented in Table 2.

Furthermore, we trained a Word2Vec model [Mikolov *et al.*, 2013] on the cleaned textual data of CoFiF and evaluated the quality of the retrieved word embeddings. Five sample word embeddings with their neighbours and the respective cosine similarity are shown in Table 3. Looking at the term *économie*, the five neighbours sample suggest that the French *économie* has important *agriculture*, *énergie* and *innovation* sectors. France was the sixth largest agricultural producer in the world and the largest within the European Union in 2011⁶. The research and innovation sector is also important to France with a total spending of 2.26% of the gross domestic product (GDP) leading to the fourth position among all the countries in the Organisation for Economic Co-operation and Development (OECD). The energy (*énergie*) sector plays its role within the French economy (*économie*); France is leading worldwide when it comes to nuclear energy and, as a re-

⁶<https://web.archive.org/web/20111009235442/http://ambafrance-us.org/spip.php?article511>

sult, the smallest emitter of carbon dioxide among the seven largest industry nations⁷. Similarly, one can identify the relatedness of terms such as *bénéfice*(gain), *profit*(profit), *versement*(payment), *solde*(balance), *résultat*(result), and *dividende*(dividend). Hence, we can say that the Word2Vec model can adequately capture the relations between terms in CoFiF.

5 Conclusion

In this paper, we present a novel corpus comprising French annual and semester reports named CoFiF. CoFiF contains a total of 188 million words and is, due to its careful company selection, a good representation for organizational writing in the French business and economic domain. Our preliminary analysis shows CoFiF’s potential to foster business, economic, and management research in the French language. Furthermore, we created two character-level language models which can be used in manifold ways such as the calculation of sentence perplexities or the extraction of word embeddings for downstream tasks. Altogether, this work aims at paving the way for further research in this area which was, until now, hindered by the absence of a publicly available language resource.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, co-funded by the European Regional Development Fund.

⁷https://unstats.un.org/unsd/environment/air_co2_emissions.htm

References

- [Abouda and Baude, 2005] Lotfi Abouda and Olivier Baude. Constituer et exploiter un grand corpus oral: choix et enjeux théoriques. le cas des eslo. In *Corpus en Lettres et Sciences sociales, Des documents numériques à l'interprétation*, 2005.
- [Akbik *et al.*, 2018] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.
- [Akbik *et al.*, 2019] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Pooled contextualized embeddings for named entity recognition. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, page to appear, 2019.
- [Barlow, 2000] Michael Barlow. *Corpus of Spoken, Professional American-English*. Rice University, 2000.
- [Benzitoun *et al.*, 2016] Christophe Benzitoun, Jeanne-Marie Debaisieux, and Henri-José Deulofeu. Le projet orféo: un corpus d'étude pour le français contemporain. *Corpus*, (15), 2016.
- [Branca-Rosoff *et al.*, 2000] Sonia Branca-Rosoff, Serge Fleury, Florence Lefevre, and Mat Pires. Discours sur la ville. *Corpus de français parlé parisien des années*, 2009, 2000.
- [Cabré, 2007] M Teresa Cabré. Constituer un corpus de textes de spécialité. *Cahiers du CIEL*, pages 37–56, 2007.
- [Candito and Seddah, 2012] Marie Candito and Djamé Seddah. Le corpus sequoia: annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical (the sequoia corpus: Syntactic annotation and use for a parser lexical domain adaptation method)[in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN*, pages 321–334, 2012.
- [Chelba *et al.*, 2013] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.
- [Content *et al.*, 1990] Alain Content, Philippe Mousty, and Monique Radeau. Brulex. une base de données lexicales informatisée pour le français écrit et parlé. *L'année Psychologique*, 90(4):551–566, 1990.
- [Daudert and Ahmadi, 2019] Tobias Daudert and Sina Ahmadi. Nuig at the finsbd task: sentence boundary detection for noisy financial pdfs in english and french. In *The First Workshop on Financial Technology and Natural Language Processing (FinNLP 2019)*, Macao, China, 2019.
- [Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [Eshkol-Taravella *et al.*, 2010] Iris Eshkol-Taravella, Olivier Baude, Denis Maurel, Linda Hriba, Céline Dugua, and Isabelle Tellier. Un grand corpus oral disponible: le corpus d'orléans 1 1968-2012. *Traitement automatique des langues*, 53(2):17–46, 2010.
- [Foltête, 1999] Isabelle Foltête. 11. analyse du discours économique dans le cadre d'une linguistique distributionnelle et transformationnelle. *Modèles linguistiques*, 20(40):119–134, 1999.
- [Gallego-Hernández, 2013] Daniel Gallego-Hernández. Comenego (corpus multilingüe de economía y negocios): a metadiscursive analysis approach. *Procedia-Social and Behavioral Sciences*, 95:146–153, 2013.
- [Gautier, 2012] Laurent Gautier. *Les discours de la bourse et de la finance*, volume 94. Frank & Timme GmbH, 2012.
- [Grabar *et al.*, 2018] Natalia Grabar, Vincent Claveau, and Clément Dalloux. CAS: French corpus with clinical cases. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 122–128, Brussels, Belgium, October 2018. Association for Computational Linguistics.
- [Guillot *et al.*, 2008] Céline Guillot, Serge Heiden, Alexei Lavrentiev, and Christiane Marchello-Nizia. Constitution et exploitation des corpus d'ancien et de moyen français. *Corpus*, (7), 2008.
- [Händschke *et al.*, 2018] Sebastian GM Händschke, Sven Buechel, Jan Goldenstein, Philipp Poschmann, Tinghui Duan, Peter Walgenbach, and Udo Hahn. A corpus of corporate annual and social responsibility reports: 280 million tokens of balanced organizational writing. In *Proceedings of the First Workshop on Economics and Natural Language Processing*, pages 20–31, 2018.
- [Kloptchenko *et al.*, 2004] Antonina Kloptchenko, Tomas Eklund, Jonas Karlsson, Barbro Back, Hannu Vanharanta, and Ari Visa. Combining data and text mining techniques for analysing financial reports. *Intelligent Systems in Accounting, Finance & Management: International Journal*, 12(1):29–41, 2004.
- [Koehn, 2005] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.
- [Kogan *et al.*, 2009] Shimon Kogan, Dimitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280. Association for Computational Linguistics, 2009.
- [Kunstmann and Stein, 2006] Pierre Kunstmann and Achim Stein. Le nouveau corpus d'amsterdam. *Le nouveau corpus d'Amsterdam. Actes de l'atelier de Lauterbad*, pages 9–27, 2006.
- [Lee *et al.*, 2014] Heeyoung Lee, Mihai Surdeanu, Bill MacCartney, and Dan Jurafsky. On the importance of text analysis for stock price prediction. In *LREC*, pages 1170–1175, 2014.

- [Leech, 1992] Geoffrey Neil Leech. 100 million words of english: the british national corpus (bnc). 1992.
- [Loper and Bird, 2002] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.
- [Mariani *et al.*, 2018] Joseph Mariani, Gil Francopoulo, Patrick Paroubek, and Frédéric Vernier. The nlp4nlp corpus (ii): 50 years of research in speech and language processing. *Frontiers in Research Metrics and Analytics*, 3:37, 2018.
- [Martineau, 2008] France Martineau. Un corpus pour l’analyse de la variation et du changement linguistique. *Corpus*, (7), 2008.
- [Merity *et al.*, 2016] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [Mondada and Pfänder, 2016] Lorenza Mondada and Stefan Pfänder. Corpus international écologique de la langue française (ciel-f): un corpus pour la recherche comparée sur le français parlé. *Corpus*, (15), 2016.
- [Paul and Baker, 1992] Douglas B Paul and Janet M Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362. Association for Computational Linguistics, 1992.
- [Peters *et al.*, 2018] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- [Verlinde, 1997] Serge Verlinde. Le vocabulaire des fluctuations dans le discours économique: synonymie et combinatoire. *Meta: journal des traducteurs/Meta: Translators’ Journal*, 42(1):5–14, 1997.
- [Vincent and Winterstein, 2013] Marc Vincent and Grégoire Winterstein. Building and exploiting a french corpus for sentiment analysis (construction et exploitation d’un corpus français pour l’analyse de sentiment)[in french]. *Proceedings of TALN 2013 (Volume 2: Short Papers)*, 2:764–771, 2013.

Step-wise Refinement Classification Approach for Enterprise Legal Litigation

Ying Mao, Xian Wang, Jianbo Tang, Changliang Li*

Kingsoft

{maoying, wangxian, tangjianbo, lichangliang}@kingsoft.com

Abstract

In the field of finance and lawsuit, data mining technology has absolute broad market prospect but also is a challenging task. Past years have witnessed great successes of data mining in finance and lawsuit related applications. Most existing work usually focus on providing litigation risk assessment and outcome prediction services for the clients. However, the research on the legal litigation type for enterprise is limited. In this paper, we focus on enterprise lawsuit category prediction and propose a novel approach to refine the problem as a classification task. First, We evaluate the possibility distribution of legal documents received by the enterprise, then distinguish the specific legal litigation type. We apply our method on *International Big Data Analysis Competition*¹ launched by *IEEE ISI Conference 2019* and scored the first place in the final leader-board.

1 Introduction

Artificial intelligence is developing very rapidly in today's society, so we hope to introduce it into the legal field. From the rule-based expert legal system (transforming legal experts' legal knowledge and experience into a computer language in the form of rules) to the autonomous system supported by deep learning, machine learning, and big data, the deeper and broader impact of artificial intelligence on the legal industries has only just begun. If we can extract valid information from the company's legal information to predict the type of received legal documents, our work can not only help the company prepare for litigation in advance, but also provide early warnings of its operating status.

Litigation/Lawsuit² is a kind of legal action, which is divided into civil and criminal categories. The former plaintiff is the victim's party, and the law is resorted to because there are unresolved disputes. The latter involves criminal offences, and the government authorities sue the suspects (prosecution). The proceedings are divided into first and second instance, and may also be final. A lawsuit may involve dispute

resolution of private law issues between individuals, business entities or non-profit organizations. The function of litigation is not limited to the discovery of historical facts that have occurred in the past, but also establishes the link between fault and responsibility, crime and punishment through the process of litigation, thereby conveying to citizens a message of how to behave and accountability.

Enterprises would receive different type of legal litigation documents for various reasons in the business process, such as tax evasion, arbitrary discharge of industrial sewage, arrears of payment, etc. Such information which can be obtained from the legal information report of the enterprise is generally historical time information. The *IEEE ISI Conference 2019* launched the *International Big Data Analysis Competition* and one of the objective aims to predict the type of legal documents that are most likely to be received by companies.

In this paper, we propose a new approach using step-wise refinement strategy to predict the top two most possible legal litigation types for each publicly traded company. In the first step, considering whether the legal documents is received by enterprise, we introduce the binary-classification model. After evaluate the possibility distribution of received legal documents, in the second step multi-class classification processing will then be performed to identify the specific legal litigation type. During these steps, the models refine more detailed information from the output produced by last step. We apply our method on *International Big Data Analysis Competition* and score the first place in the final leader-board. As it turns out, achieving the remarkable result reveals our step-wise refinement approach significantly outperforms state-of-the-art techniques in legal litigation prediction field.

2 Related Work

The traditional solution to the task of legal type prediction is manual judgment, which means experienced professionals draw conclusions based on relevant data. The main analysis process is generally based on the analysis of the number, regional distribution, and subject matter of the litigation cases that the company participates as a litigation participant. Through analyse the corresponding data, the main types of litigation cases of the corresponding enterprises in the earlier period of the data acquisition period can be obtained with a high probability.

*the corresponding author

¹<http://www.linkx.ac.cn/#/title>

²<https://en.wikipedia.org/wiki/Lawsuit>

The first part is the analysis of the litigation case types in the history of the company. The major categories of litigation cases are generally criminal, administrative and civil. In fact, most enterprises is main body of operation in our country and their main business activities should be based on business behaviors of civil and commercial activities, so the development of disputes and types of litigation are mostly civil and commercial litigation. When civil cases are further subdivided, they can be distinguished by referring to the provisions of the *Supreme People's Court* on civil cases. The civil case generally represents the problems existing in the management methods or business concepts, and may also reflect its operating status and some industry characteristics to a certain extent. If there are a large number of cases of product quality disputes, it indicates that the company may have flaws in the process of product quality control. If there are a large number of labor dispute cases, it indicates that the company may have problems such as daily violation of labor contract law management and large mobility of industry personnel due to poor management. If an enterprise has an administrative case or a criminal case, there are serious illegal activities on behalf of the company's production and management behavior, such as meat product enterprises that have been administratively recalled due to quality problems, and vaccine companies that have been investigated by criminals. Once this type occurs and a case with extensive public opinion influence is formed, it will often have a greater impact on the sustainable operation of the enterprise. It may also result in a series of civil cases, such as quality claims, personal injury claims, shareholder lawsuits, etc. It can be considered that if there is no major change in the management or strategy of the company, in addition to the impact of the phased factors, the above situation may have a certain continuity, thus which also can help to predict the trend of the company's frequent cases.

The second part is the analysis of the area where the case occurred and the location of the main business of the company. The occurrence area may reflect the market share of the business operations in the relevant regions. Generally speaking, the outbreak of the case has a certain positive correlation with the market share of the corresponding region. The larger the business volume, the more the problem will be. Sometimes it can also indicate that there is a problem with the local management and operation of the company, which leads to a concentrated outbreak of controversy.

The third part is the analysis of the subject matter. Generally speaking, in the relevant cases, the greater the type of the subject, the stronger the impact on the business operation. It means that the management problems of the relevant parties may be more prominent.

3 Task Definition

The *International Big Data Analysis Competition* subject aims to predict the top two most possible legal litigation types for each publicly traded company according to companies' full size information.

For this competition, the organizer provides 18 data forms³, including multi-dimensional information of publicly

³<https://pan.baidu.com/s/1YaJcn-lpWVN2TSFDUFLwYw>

traded companies in the past years, which can be seen from Figure. 1. These data are from official statistic platform. There is one unique ID for each company.

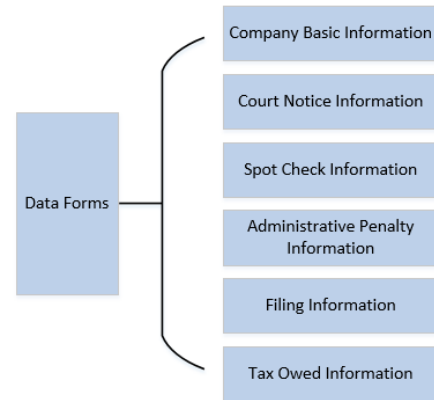


Figure 1: Data forms

The organizer has anonymized, discretized, and normalized some of the information for the protection of data such as corporate secrets, intellectual property rights, and debt information. The court notice is the core form, and the organizer has anonymized the types of legal documents in this form. The types of legal documents processed are expressed as: A-Q and others. As shown in Figure. 2, there is a wide disparity in the amount of different types. We need to extract the top two most possible legal litigation types for each company from this data form.

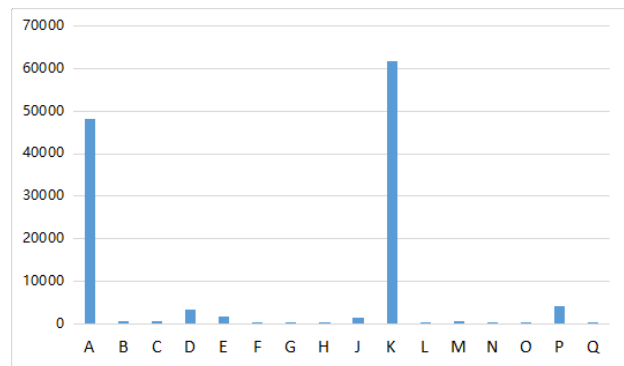


Figure 2: Distribution of the legal litigation type

They provide 3,000 companies as training data sets (actually about 1,500, with about 700 effective data). The company name is anonymized and expressed by enterprise number between 1001 and 4000. The data in some of the tables is incomplete, and the companies provided may not all be used as training sets, and flexible data processing is required. Also the organizer provides about 500 companies as test data sets (actually about 200) with enterprise number between 4001 and 4500. We should calculate the types of legal documents that these companies may receive the most and the second most.

4 Approach

In this section, we will present end-to-end procedures of our strategy. This work consists of four different parts which are preprocessing, feature engineering, model establishment and ensemble. The overall procedure is illustrated in Figure 3.

We found that splitting the original problem into three sub-problems can help to better fulfil the forecasting task. The three sub-problems aim to evaluate whether the enterprise has received the legal documents, the type of legal document that the enterprise receives the most, and the type of legal document that the enterprise receives the second most. The training data required for the three sub-problems are not the same, moreover the three will affect each other. Therefore, the first strategy aims to train the model separately, and then combining the three classification algorithms, according to the results.

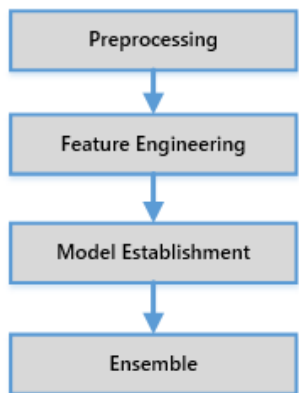


Figure 3: Overall procedure

4.1 Preprocessing

The basic principles of preprocessing for our raw data consist of five parts. Firstly, all data have to be combed on multiple time granularity, for instance we not only count the number of civil cases received by the company in the last year but also in the last two years etc. Secondly, if there are several different time stamp for same data, they should be sorted out specifically. Thirdly, the original data need to be eliminated redundancy before it is counted. Fourthly, there are plenty of data related to capital provided by organizer which need to be reprocessed carefully. For example unified unit of money, count stocks and cash respectively. Fifthly, in order to handle the missing values problem we should attempt to use different strategies as much as possible such as mean, mode, median and so on.

Simplifying prediction labels is one of our special treatment. As shown in Figure 4, we find there are unbalanced distribution of legal litigation type. The sum of proportion of type A, D and K is over 90%, on the contrary some labels like type C, F, O and Q are less than 1%. To deal with this situation, some labels have been considered as exception value if their proportion are less than 1% and some labels have been combined into one if their proportion are larger than 1% and

less than 5%. The same strategy is applied to organize the second most legal litigation type data.

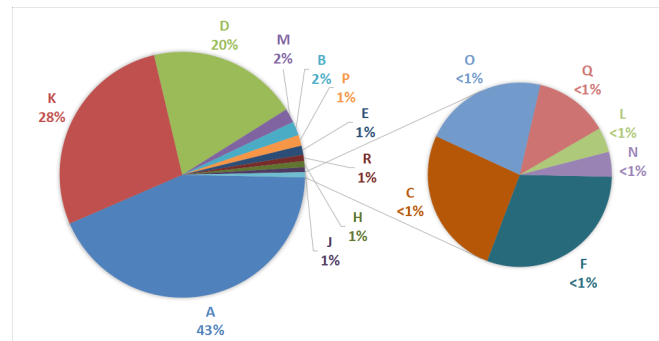


Figure 4: Distribution of the most legal litigation type

Another special treatment we have applied on the raw data is using external knowledge. When we simplify the information about administrative penalty, using the external knowledge make the entire preprocessing process more rational. In the raw data, the descriptions about context of administrative penalty for company every year are different and complicated. After reviewing the relevant regulations on administrative punishment, we make a summary of the descriptions about administrative punishment. Extracting important information from the complicated description, and based on this we roughly divided them into new categories, such as commodity trademark related, computer infringement related, illegal occupation related and so on.

As shown in Figure 5, the presence of outliers in the data sets can be easily confirmed. Dealing with outliers is also a very important part of data preprocessing. The key to this procedure is eliminate outliers while keeping adequate data volume. Because the amount of relevant data provided by the official is limited, it is very important to be careful when choosing the right method to ensure that do not eliminate too much data. By comparing several different common methods, we choose to detect outliers by artificial experience and PauTa Criterion principle[Li *et al.*, 2016], and at the end the problem of outliers has been solved partly.

4.2 Feature Engineering

After preprocessing, some important features which are not mentioned in the official material need to be constructed. There are three methods have been used to construct the features, basically the construction of features is designed by data characteristics.

First, considering that the growth rate of important indicators in the short term can predict the company's changes in a certain area, trend features have been made by assessing the growth of data in the current two years relative to the company's previous data. Such as for the data sets of the judicial documents, the work to calculate the average number of civil cases of each company in 2016-2017 and the number of civil cases in 2015 needed to be done simultaneously. Soon afterwards, using the previously obtained data, the growth rate of civil cases can be calculated from the period of 2016 to the end of 2017 relative to the previous year.

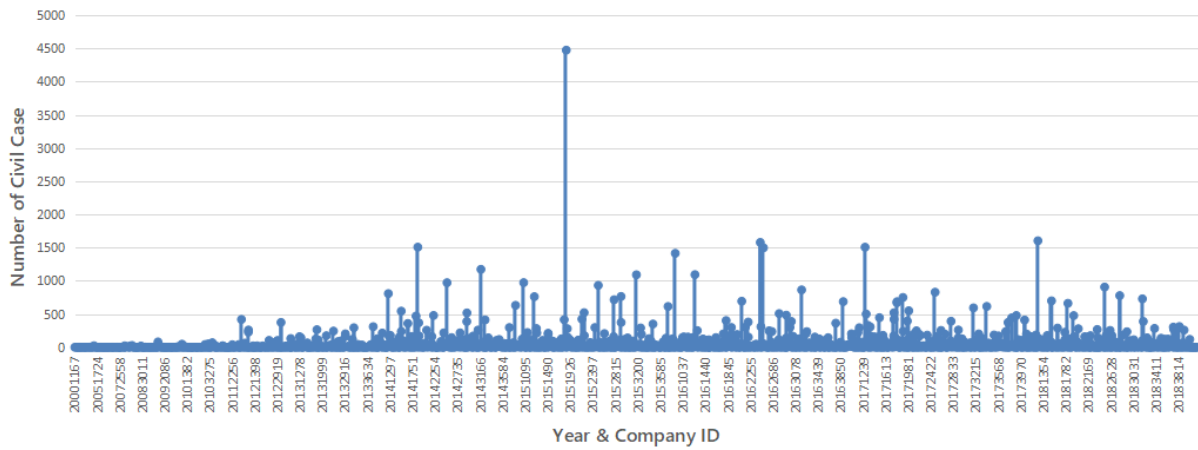


Figure 5: Distribution of number of civil case

Second, considering the differences between regional policies and the problems and challenges faced by various industries, statistical features need to be created to improve system classification capabilities. The main method for constructing statistical features is count the ranking information between different regions, cities, industries, and sub-sectors as well as the proportion of prediction labels, according to the average registered capital and other information. The details are as shown in Figure. 6, there is significant difference between industries for whether the legal documents will be received. Based on empirical analysis, our team believes that the risks of receiving legal documents in diverse industries are different. The traditional industries with mature product quality control and standardized personnel management procedures are less threatened by legal documents, on the other hand new industries are still at the stage of exploration, management is not standardized, and legal instruments are more likely to be received. Similarly, the difference of scale of enterprise may also cause different preferences of legal litigation type. This is also why we construct the feature for describing enterprise scale by using registered capital and number of employees.

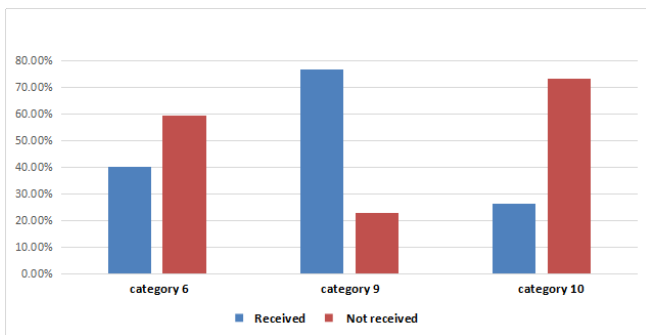


Figure 6: Distribution of whether to receive legal litigation in different industry category

Third, due to constructing polynomial features is one of the common methods for constructing new features, besides it is widely used in statistical models to explore the effect

of compound variables on y. Based on experience, we use second-order polynomial features to enrich the feature sets.

4.3 Model Establishment

Through consulting literature and investigation, four models enter the candidate program which are LightGBM[Ke *et al.*, 2017], Xgboost[Chen and Guestrin, 2016], RandomForest[Svetnik *et al.*, 2003] and Neural Network[Hansen and Salamon, 1990]. After experimental verification, the results show that LightGBM, XGBoost work best for this problem.

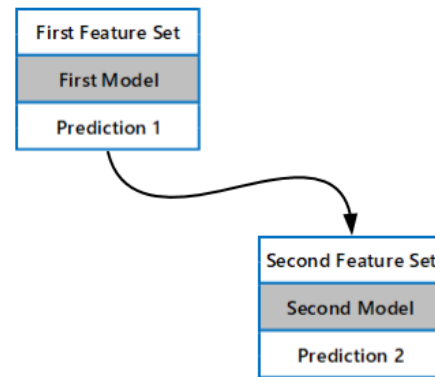


Figure 7: Single classification model for multi-class

To better solve the classification problem, a novel step-wise refinement strategy is designed. First, the problem is composed of one binary classification problem and two multi-class classification problems. The binary classification model is used to predict whether a company will receive legal documents over a period of time by LightGBM model. The multi-class classification models are used to predict the specific legal litigation type by LightGBM and Xgboost models. The detailed procedure for multi-class classification is to predict the type of legal litigation that the company will receive the most firstly, then the predicted result need to be added as a

feature to the input feature sets of the second model. At the end, the second model is used to predict the type of legal litigation that the company may receive the second most. The final results are derived from the binary classification results and the multi-class classification results. The details are shown in Figure. 7.

For the LightGBM and XGBoost models of the multi-class classification problem, 10 models are constructed by 10 fold cross-validation, and the final prediction value is obtained by the voting mechanism as the single model final prediction result. This action improves model accuracy and reduces overfitting. The details are shown in Figure. 8.

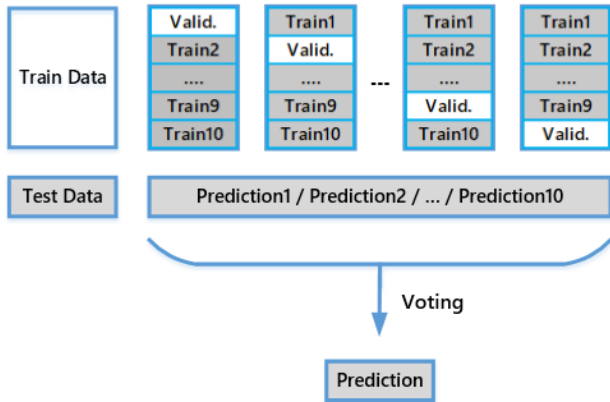


Figure 8: 10-fold cross validation

One challenge is that LightGBM and XGBoost models contain some super-parameters, and the selection of parameters have a certain impact on the classification results. Commonly used parameter search methods include: Random Search[Bergstra and Bengio, 2012], Grid Search and Bayesian Search[Feurer *et al.*, 2015]. After many attempts, Bayesian search is finally selected due to its superior performance.

4.4 Ensemble

In real-world cases, training models to generalize on datasets can be a very challenging problem as it can contain many underlying distributions. Certain models will do well in modelling one aspect of this data while others will do well in modelling the others. Ensemble provides a solution where we can train these models and make a joint prediction where the final accuracy is better than each of the individual models.⁴

There are several commonly used ensemble methods, such as voting[Rokach, 2010], averaging, and stacked generalization[Williams and Gong, 2014] and blending. After many experiments and verification applied in the multi-class classification model task, voting is finally adopted. As shown in Figure. 9, in order to ensure the effect of model ensemble, there are differences in the input feature sets of the four multi-class classification models. All of the feature sets

which are used are selected based on their performance in single model.

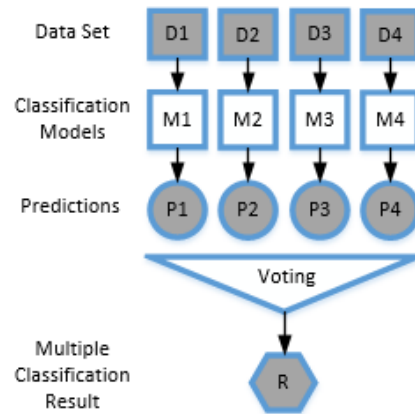


Figure 9: Model ensemble

As shown in Figure. 10, our final result is coupled by the prediction results of the binary classification model and all the multi-class classification models.

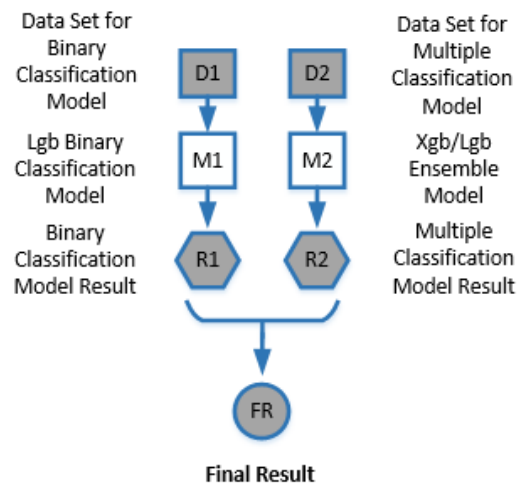


Figure 10: Combination of binary model and multiple model

The final strategy we have adopted when merging the results of the binary classification result and multi-class classification result can be summarized as: firstly, if the prediction result of binary classifications is positive, then focus on the multi-class classification result, otherwise it is classified as negative. Secondly, if the multi-class classification result is displayed as others which is created when we have merged some labels, further prediction is performed according to the distribution of the original data sets, otherwise it is recorded as the original multi-class classifier result.

5 Experiments

In this section, the final result will be presented. The experiment environment we used is Python 3.6 and calls tools such

⁴<https://medium.com/weightsandbiases/an-introduction-to-model-ensembling-63effc2ca4b3>

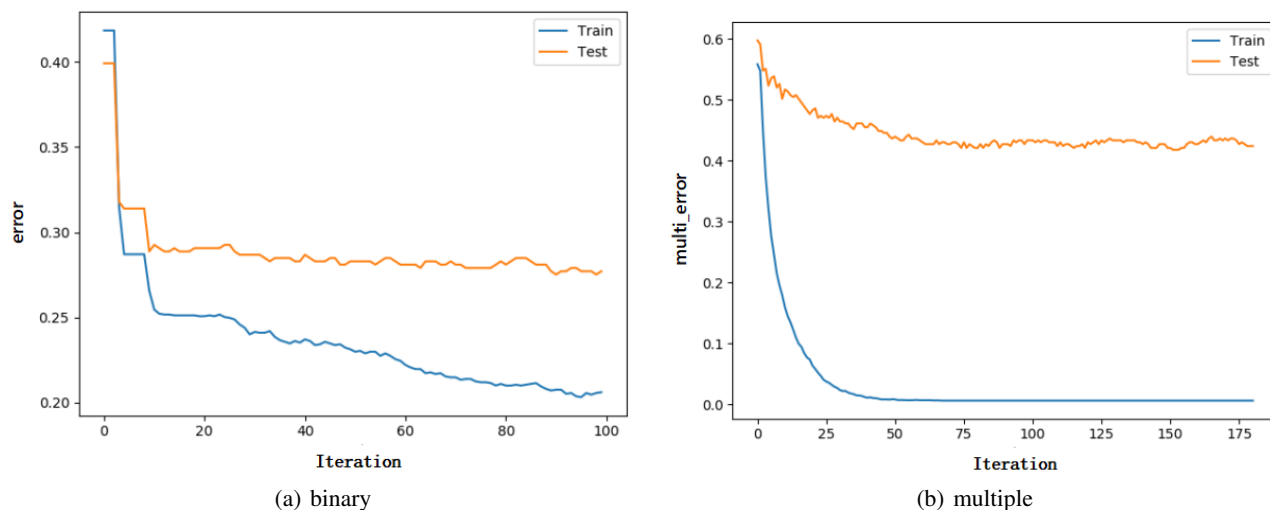


Figure 11: Training and testing curve for classification

as pandas 0.23.0, sklearn 0.19.1, lightgbm 2.2.3, etc.

After data preprocessing, the data volume of the multi-class classification model is about 3200. On the contrary, the input volume for binary model is more than 4,700 and the validation set data volume is 133. The number of features for binary model and four multi-class model are all around 20 to 25. Negative samples of relatively long time (such as 2011-2012, 2012-2013) can not fully indicate that the company did not receive legal documents during the year, but because of the loss of information and the incomplete statistics. By counting the proportion of positive and negative samples from 2011 to 2018, our team found that the proportions fluctuated significantly, which clearly confirmed our conjecture. Therefore, in order to eliminate the influence of observation errors of these historical data, we only used the negative samples of 2017-2018 to join the training set.

The training/testing curve for binary and multiple classification tasks shows in Figure. 11. The two figures show that the prediction accuracy of the binary classification model as well as multiple classification models all converge to a high level.

The organizer provides an evaluation index algorithm for the model effect. For every enterprise, if the model predicts the correct results for both the most legal litigation type and the second most legal litigation type, then the score for this enterprise should be the max score, which is 100. If the model predicts the correct result for the most legal litigation type but the wrong result for the second most legal litigation type, then the score for this enterprise should be a small amount of max score, for example 35. The final score of the test datasets should be the average score of all the enterprises.

Finally we scored 44 points and achieved the first place in the competition.⁵ The excellent score of the competition also shows the superiority of the model in this legal litigation type prediction field.

⁵<http://www.linkx.ac.cn/#/ranking>

6 Conclusion

In this paper, we presented a step-wise refinement classification approach for legal litigation prediction which combines binary and multi-class classification model based on parameter search and ensemble methods. The innovation for this design is to evaluate the possibility distribution of legal documents received by enterprises before further distinguish the specific type of legal litigation. When evaluated on officially provided datasets, our model significantly outperforms all the counterparts.

While our approach provides substantial performance gain, there is still room for improvement. In the future, we would like to discover more distinguishing features and illustrate the interpretation of the proposed model.

References

- [Bergstra and Bengio, 2012] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.
- [Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.
- [Feurer *et al.*, 2015] Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [Hansen and Salamon, 1990] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10):993–1001, 1990.
- [Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and

- Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.
- [Li *et al.*, 2016] Limin Li, Zongzhou Wen, and Zhongsheng Wang. Outlier detection and correction during the process of groundwater level monitoring based on pauta criterion with self-learning and smooth processing. In *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems*, pages 497–503. Springer, 2016.
- [Rokach, 2010] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010.
- [Svetnik *et al.*, 2003] Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6):1947–1958, 2003.
- [Williams and Gong, 2014] Trefor P Williams and Jie Gong. Predicting construction cost overruns using text mining, numerical data and ensemble classifiers. *Automation in Construction*, 43:23–29, 2014.

CoSACT: A Collaborative Tool for Fine-Grained Sentiment Annotation and Consolidation of Text

Tobias Daudert^{1*}, Manel Zarrouk¹ and Brian Davis²

¹Insight Centre for Data Analytics, National University of Ireland, Galway

²Department of Computer Science, Maynooth University

{tobias.daudert, manel.zarrouk, brian.davis}@insight-centre.org

Abstract

Recently, machine learning and in particular deep neural network approaches have become progressively popular, playing an increasingly important role in the financial domain. This results in an increased demand for large volumes of high quality labeled data for training and testing. While annotation tools are available to support text analysis tasks such as entity recognition and sentiment classification for generic content, there is an absence of annotation tools purposely built for the financial domain. Hand in hand with this, there are also no existing best practices for sentiment annotation in the financial domain. To address this issue, we suggest fundamental practices for the creation of new datasets for this domain and integrate these into our annotation tool. We present CoSACT, a server-based tool which supports the collaborative annotation and consolidation of a dataset purposely built for the financial domain.

1 Introduction

The increase in popularity and adoption of social media in recent years has generated a deluge of valuable, high volume and volatile user-generated content [Derczynski *et al.*, 2015]. Microblogs have become a central medium of communication and are used in almost all areas of our daily life. They are mainly characterized by their short length, nonetheless, they can be rich in content and highly opinionated [Sinha, 2014]. Microblogs are used by services such as Stocktwits¹, Twitter², or Facebook³. The growth of Twitter, by 1,006% between 2010 and 2015, clearly shows the high importance of short messages.⁴ Opinion Mining from short texts, such as microblogs, has become an increasingly important task [Derczynski *et al.*, 2015; Derczynski *et al.*, 2013]. Data generated from online communication acts as a potential goldmine for discovering knowledge [Dey and Haque,

2009]. In the area of finance, sentiment acquired from text can positively influence businesses in many ways. For example, it can be used as an indicator for trading or provide knowledge about the public perception of a company. Sentiment classification and opinion mining tools increasingly employ machine learning approaches [Missen *et al.*, 2013; Pang and Lee, 2008] requiring access to labeled data for training and testing for benchmarking learning systems. The annotation and subsequent consolidation of microblogs into a high-quality gold standard is a challenging, time-consuming and labor-intensive task. Annotation tools aim to reduce the burden associated with this task, however, there are few publicly available annotation tools dedicated to capturing the sentiment in short texts [Trakultaweekoon and Klaithin, 2016] and often, unlike our tool, they do not focus deeper beyond the "document level" towards fine-grained annotation of sentiment annotation at the entity/target level.

In this paper, we present CoSACT⁵, the first annotation tool incorporating a consolidation mode purposely built for the financial domain.

2 Annotation Design

Sentiment in the financial area is driven by multiple factors, hence, the quality and information content of datasets is crucial to the success of sentiment analysis. The results of sentiment analysis can have significant implications in the financial domain, positively as well as negatively. One example is algorithmic trading; on one hand, it can be very profitable and some companies (e.g. Renaissance Technologies⁶) are very successful with the computational analysis, on the other hand, algorithmic trading can lead to big losses when computers misinterpret information. Incidents such as the Lululemon Athletica Inc. case highlight this risk; their stocks dropped due to problems with the company's product - sport trousers - which was sarcastically discussed on Twitter. Automated trading systems misinterpreted these tweets as positive and long-investors lost money⁷. The more diverse information is available with a high level of detail, the better a classifier

*Contact Author

¹<https://stocktwits.com/>

²<https://twitter.com/>

³<https://www.facebook.com/>

⁴<https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>

⁵<https://github.com/TDaudert/CoSACT>

⁶<https://www.rentec.com>

⁷<https://www.theepochtimes.com/how-hedge-funds-use-twitter-to-gain-an-edge-in-trading-2227349.html>

is able to learn the underlying data characteristics. Prior to developing our annotation tool, we considered which short financial text parameters are necessary to conduct high-quality sentiment analysis; hence, our tool was developed keeping this in mind. We detail these characteristics below:

1. **Sentiment granularity** - Currently, the majority of sentiment datasets is annotated in a categorical fashion with polarity (positive, negative, neutral) [Saif *et al.*, 2013]. Although this simplifies the annotation process and leads to higher levels of inter-annotator agreements, a sentiment classifier trained on a polarity dataset will not be able to learn how to classify data with a higher granularity (i.e. in more than 3 classes). However, in the financial domain, this poses a severe drawback since the ability to capture the degree to which one text is more positive/negative than another is highly desirable. Considering a case in which one text is strongly positive and another one slightly negative, the question arises which text has a stronger impact on an asset price. This case applies to all periods in which multiple data with different sentiments exist. Comparing the sentiment polarity (positive, negative, neutral) with trading actions (buy, sell, hold), polarity classification seems to be an obvious choice, however, in the real world, there is seldom a scenario with only one data artifact to be classified. To address this, data annotated with CoSACT receives a continuous sentiment score between -1 and +1 with 0 as neutral (e.g. 0.472). However the tool has the ability to later divide the annotated data categorically into classes without limiting the granularity at the time of annotation.
2. **Multi-entity annotation** - Sentiment at document-level may not be sufficiently accurate when dealing with scenarios requiring the consideration of single entities. Many documents (i.e. tweets, posts) mention more than one entity, hence, a good dataset should aim to include one sentiment annotation for each contained entity. The benefit here is at hand; some texts might contain positive sentiment towards some entities and neutral or negative towards others. To accommodate this, each cashtag in CoSACT achieves a sentiment annotation. As you can see in Figure 1, the displayed text contains two entities represented by \$DD and \$MON; both receive an individual sentiment score.
3. **Spans** - In addition to specifying a sentiment for each entity, CoSACT also requests the span in which this sentiment is contained. This feature provides high-quality information for a sentiment classifier since this fine-grained annotation allow us to aim for an entity-level classification (i.e. determine one sentiment score for each entity in a text).
4. **Contextual dependency** - A contextual dependency is represented by additional information (i.e. image, url) apart from the text the sentiment is referring to. From the sentiment analysis point of view, this is relevant as not all information a sentiment is based on can be retrieved solely from the annotated text. A more sophisticated sentiment classifier may wish to leverage the additional external contextual information.

5. **Justification** - To facilitate the consolidation, CoSACT collects the annotator’s reasoning behind his/her annotation. Moreover, the automatic evaluation of the justifications can be used for the generation of a confidence score which provides further information to a sentiment classifier. It becomes possible to give more importance to some annotations than to those the annotator was unsure of his judgment.

3 The Tool: CoSACT

CoSACT is a server-based tool, which requires only a browser (e.g. Chrome⁸, Firefox⁹). It is developed using JavaScript¹⁰ (particularly NodeJS¹¹), a SQL database¹², HTML¹³ and CSS¹⁴. As it is utilizing a server-client architecture, multiple people can collaborate online in the annotation of the same dataset, at any point in time, without requiring the storage of large amounts of data on their local machines. It also gives the flexibility to outsource the annotation task as only a browser is required and the annotators will never come in contact with the raw dataset itself. The access to the tool and, hence, the data, is protected by a user-name and password combination. Therefore, the tool’s availability on the web carries a reduced security risk.

CoSACT evolved from two separate tools, an annotation and a consolidation tool, which were tested in real-world scenarios. These were employed in the Social Sentiment Indexes Project (SSIX)¹⁵ to create gold standards in different languages to monitor sentiment related to the financial markets. Additionally, the data used in the Semeval 2017 Task 5 competition on *Fine-Grained Sentiment Analysis on Financial Microblogs and News* was generated with this annotation and consolidation tool [Cortis *et al.*, 2017]. The gold standard dataset used by Byrne *et al.* [2016] to predict the outcome of the Brexit poll based on tweets was also created using CoSACT.

To better detail the tool’s interface, we will use numbers enclosed in parenthesis, e.g. (1), which point to specific locations in the following figures.

3.1 The Annotation Mode

The CoSACT interface for the annotation mode is shown in Figure 1. The microblog text as presented to the annotator is represented in box (3). Right to it, (4) highlights the different submission options. Here, the annotator can choose between submitting his/her annotation (green button), labeling the presented microblog as spam (red button), labeling it as irrelevant (blue button), or the annotator can click on one of

⁸<https://www.google.com/chrome/index.html>

⁹<https://www.mozilla.org/de/firefox/>

¹⁰<https://en.wikipedia.org/wiki/JavaScript>

¹¹<https://nodejs.org/en/>

¹²<https://en.wikipedia.org/wiki/SQL>

¹³<https://en.wikipedia.org/wiki/HTML>

¹⁴https://en.wikipedia.org/wiki/Cascading_Style_Sheets

¹⁵<https://ssix-project.eu/>



Figure 1: The interface of CoSACT in the annotation mode.

the two grey buttons. In this case, the annotator then chooses to label the data as “I don’t know”, in case he/she is unsure about the correct annotation, or as “Not enough information”, in case the uncertainty clearly derives from the presented data itself. This differentiation provides insights about the information content of the data as it becomes clear whether a text might have been too concise for a good interpretation or the annotator has simply not been able to interpret it. Below the presented tweet, (5) labels a field aiming at storing a justification for the annotators choice. This justification might provide relevant value in addition to the previously assigned feature. On its right, the two boxes (6) “Image” and “External” (e.g. URL) can be ticked in case the presented message is based on additional information and the content alone does not represent the sentiment fully. Location (10) in Figure 1 is the sentiment slider which the annotators use to assign a continuous sentiment between -1 (Negative) and +1 (Positive) to an entity. The text area at location (11), allows the annotators to assign one or multiple spans to the sentiment and, therefore, to specify the part of the microblog which contains the sentiment. In case an annotator is unable to assign a span, the box on top of the text area can be ticked. Finally, the the text area (12) in Figure 1 presents the selected spans; the button above is for resetting spans in case an annotator wants to revise his/her selection.

Note that the master selectors presented with a cyan background in (7)-(9) of Figure 1, affect all the other fields. For example, a change in the slider (7) is also moving the two sliders below. This functionality gives the annotators the capability to select spans for microblogs with multiple entities without the necessity to select them independently.

We chose to use a slider to determine the sentiment (See

(7)/(10) in Figure 1) to prevent the annotators from using some sentiments disproportionately. If the sentiment range would present classes, annotators would tend to move the slider towards one of the categories (e.g. 1,2,3). Similarly, annotators requested to give numbers for a sentiment might be unable to go into a high level of detail and would tend to annotate with rounded numbers (e.g. -1.0, -0.5, 0.0, 0.5, 1.0).

3.2 The Consolidation Mode

In consolidation mode, CoSACT is able to smoothly consolidate a previously annotated dataset. At the first time the user tasked with consolidation logs in, he/she is requested to specify the number of required annotations per microblog as well as to set an acceptable deviation. The deviation defines the degree of variation between the annotations of a text. These two values are then used to automatically consolidate all microblogs which fulfill the given criteria (Figure 2). A manual consolidation is required for the remaining microblogs; automatic consolidations can be amended by the consolidator. As shown in Figure 3, the example does not meet the criteria thus, it needs to be consolidated. In box 5, microblogs to be consolidated have a gray background, auto-consolidated ones are highlighted cyan, and consolidated microblogs are shown in blue. The microblog currently in use is underlaid in green. The consolidator can use the three buttons in (4) to either classify a microblog as spam, irrelevant or to save his/her consolidation. Location (6) shows the scores from the annotators. The “ALL” area (7)-(8) is similar to areas (7)-(9) of the annotation mode, as well as the mechanism of assigning a sentiment and span. The mechanism of assigning a sentiment and spans is similar to the process described in Section 3.1. The consolidator is requested to set the final sentiment as well

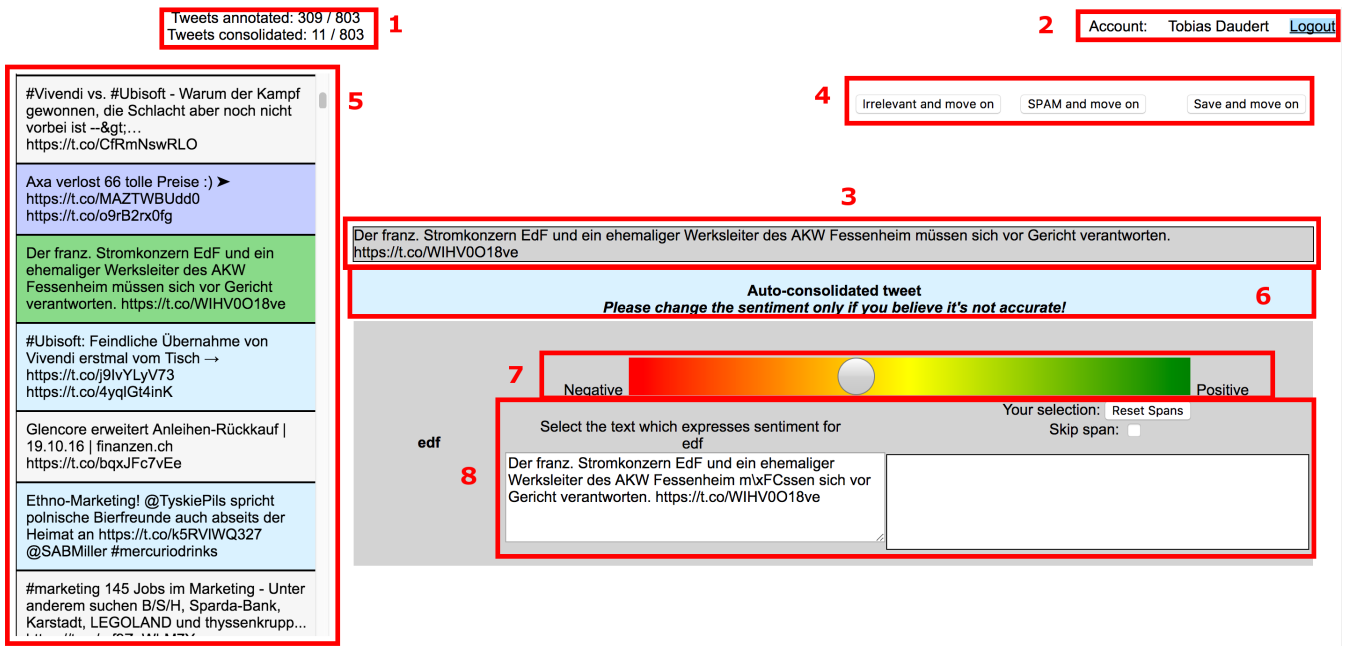


Figure 2: The interface of CoSACT in the consolidation mode presenting an auto-consolidated microblog.



Figure 3: The interface of CoSACT in the consolidation mode presenting a microblog to be consolidated.

as to set the spans, all under consideration of the annotations given and shown.

4 Related Work

Due to the current high interest in microblogs and sentiment analysis, tools focusing on sentiment annotation of short text have become of relevant importance. Some examples of ser-

vice based general purpose collaborative linguistic and semantic annotation tools include WebAnno [de Castilho *et al.*, 2014], GATE Teamware [Bontcheva *et al.*, 2013] which is integrated into the AnnoMarket Text Mining Service [Tablan *et al.*, 2013], TURKSENT [Eryigit *et al.*, 2013], and SenseTag [Trakultaweekoon and Klaithin, 2016]. Due to our interest in sentiment annotation, we focus on two of these publicly available tools which also focus on this particular task: TURKSENT and SenseTag. However, it is important to clarify that these tools are not purposely built for the financial domain nor were applied in any financial context. We chose to address these, as they are the most closely related tools to our work.

Eryigit [2013] were the first to create a publicly available sentiment annotation tool. It incorporates a manual or semi-automatic linguistic annotation layer which includes text normalization, named entity recognition, morphology, and syntax tasks. TURKSENT provides a software as a service which does not require a specific platform, and it is also accessible by multiple users. For the annotation, categorical labels supported by images (i.e. emoticons) are used; these range from 1-3 and include an additional ambivalent label. Two tasks are assigned during the annotation process. In the first task, the annotator is required to provide a general sentiment label for the text and select an appropriate comment category and sentence type. For the second step, it is performed a target based annotation where tuples of the brand, product/service and features are provided. The second analysed tool, SenseTag, consists of three components: data collection, data annotation and administrative operation. The data is automatically collected from microblogs and classified into four domains. This information is then pre-processed and stored in a database. Non-annotated messages are selected randomly and are then manually tagged for each word. The words are classified into 4 categories: positive, negative, feature, and entity. Finally, the last component allows the management of the tags, and the tagging tool as well as data handling.

Considering these tools, we identified generic tool characteristics and shortcomings, which we address with CoSACT:

1. **Complex pre-processing.** While these tools rely on an additional annotator task to deal with entity recognition, CoSACT applies a simplistic approach focusing on a previous input of entities or regular expressions (i.e. regex) to extract them. Either the entities to be found are being specified beforehand, or a regex is given to identify them. This removes the additional laborious task for the annotators. Our tool accepts free text, hence, not demanding the pre-identification of categories or features.
2. **Categorical sentiment.** Both TURKSENSE and SenseTag utilize a small, categorical range of sentiment. In CoSACT, we use a wide, continuous span from -1 to +1, which allows for the extraction of a fine-grained sentiment score. CoSACT is the only tool providing a fine-grained sentiment scale.
3. **Overall score.** Our tool provides a sentiment score for the text, as a whole, and for the entities (e.g. \$Volkswagen) and/or tickers (e.g. VOW). This is not present in SenseTag.

4. **Contextual Dependencies.** Short messages may contain additional information conveyed by an image or URL. CoSACT takes this into account by allowing the user to identify messages where this situation occurs.
5. **Consolidation.** Our tool implements a consolidation mode which allows a user, preferentially a domain expert, to consolidate already annotated datasets. This is useful in cases where the minimum number of annotations is not met and/or when the deviation is exceeded, in the remaining scenarios the microblog is auto-consolidated. This is a vital step for the development of quality gold standards and not yet implemented in other tools.
6. **Usability features.** CoSACT's interface is able to indicate the number of messages in the annotation and consolidation tasks and the overall progress. It also has the advantage of easily allowing the user to review already consolidated text. Annotated text cannot be reviewed, this is a deliberate design feature. We believe the annotator should focus on the current text and not review the assigned sentiment in accordance with other messages seen before.

5 Conclusion

In this paper, we addressed the growing need for quality labeled data for training supervised sentiment classification tools for the financial domain. We established fundamental practices we believe to be essential to improve the dataset quality and incorporated them in an annotation and consolidation tool designed for the financial domain. Although collecting many different parameters from an annotator, CoSACT simplifies the process of annotation by providing an automatic consolidation as well as a manual consolidation interface. It combines the benefits of collecting a magnitude of information while keeping the manual effort low. Entities can be annotated with a sentiment, one or multiple spans expressing it, a justification for the annotation, and contextual information. In addition, data can be classified as spam or irrelevant and annotators can click on "I don't know" in case they are overwhelmed, or can mark texts with "Not enough information" in case they perceive it as not revealing enough information. All these parameters contribute meaningfully to automated sentiment analysis in a financial setting. Since this tool is server-based, it only needs to be set up once and allows annotators and consolidators to work on data with a minimum of effort, requiring not more than a standard browser. Our tool encourages the efficient engineering of quality fine-grained labeled datasets for developing, testing and benchmarking (semi-)supervised learning systems for entity oriented sentiment analysis.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, co-funded by the European Regional Development Fund.

References

- [Bontcheva *et al.*, 2013] Kalina Bontcheva, Hamish Cunningham, Ian Roberts, Angus Roberts, Valentin Tablan, Niraj Aswani, and Genevieve Gorrell. Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029, 2013.
- [Byrne *et al.*, 2016] David Byrne, Angelo Cavallini, Ross McDermott, Manuela Hürlimann, Frederico Caroli, Malek Ben Khaled, André Freitas, Manel Zarrouk, Laurentiu Vasiliu, Brian Davis, et al. In or out? real-time monitoring of brexit sentiment on twitter. *SEMANTiCS 2016*, 2016.
- [Cortis *et al.*, 2017] Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 519–535, 2017.
- [de Castilho *et al.*, 2014] Richard Eckart de Castilho, Chris Biemann, Iryna Gurevych, and Seid Muhie Yimam. Webanno: a flexible, web-based annotation tool for clarin. In *Proceedings of the CLARIN Annual Conference (CAC)*, 2014.
- [Derczynski *et al.*, 2013] Leon RA Derczynski, Bin Yang, and Christian S Jensen. Towards context-aware search and analysis on social media data. In *Proceedings of the 16th international conference on extending database technology*, pages 137–142. ACM, 2013.
- [Derczynski *et al.*, 2015] Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49, 2015.
- [Dey and Haque, 2009] Lipika Dey and Sk Mirajul Haque. Opinion mining from noisy text data. *International Journal on Document Analysis and Recognition*, 12(3):205–226, 2009.
- [Eryigit *et al.*, 2013] Gülsen Eryigit, Fatih Samet Cetin, Meltem Yanik, Tanel Temel, and Ilyas Çiçekli. Turksent: A sentiment annotation tool for social media. In *LAW@ACL*, pages 131–134, 2013.
- [Missen *et al.*, 2013] Malik Muhammad Saad Missen, Mohand Boughanem, and Guillaume Cabanac. Opinion mining: reviewed from word to document level. *Social Network Analysis and Mining*, 3(1):107–125, 2013.
- [Pang and Lee, 2008] Bo Pang and Lillian Lee. Opinion Mining and Sentiment Analysis Bo. *Foundations and Trends® in Information Retrieval*, 2(1-2):1–135, 2008.
- [Saif *et al.*, 2013] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold. 2013.
- [Sinha, 2014] Nitish Sinha. Using big data in finance : Example of sentiment extraction from news articles. Technical report, Board of Governors of the Federal Reserve System (US), 2014.
- [Tablan *et al.*, 2013] Valentin Tablan, Kalina Bontcheva, Ian Roberts, Hamish Cunningham, Marin Dimitrov, and AD Ontotext. Annomarket: An open cloud platform for nlp. In *ACL (Conference System Demonstrations)*, pages 19–24, 2013.
- [Trakultaweekoon and Klaithin, 2016] Kanokorn Trakultaweekoon and Supon Klaithin. Sensetag: A tagging tool for constructing thai sentiment lexicon. In *Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on*, pages 1–4. IEEE, 2016.

Financial Text Data Analytics Framework for Business Confidence Indices and Inter-Industry Relations

Hiroki Sakaji¹, Ryota Kuramoto¹, Hiroyasu Matsushima¹, Kiyoshi Izumi¹,
Takashi Shimada¹ and Keita Sunakawa³

¹School of Engineering, The University of Tokyo

²The Bank of Okinawa

{sakaji, matsushima, izumi}@sys.t.u-tokyo.ac.jp, m2017rkuramoto@socsim.org,
houjin-kikaku3@okinawa-bank.co.jp

Abstract

In this paper, we propose a novel framework for analyzing inter-industry relations using the contact histories of local banks. Contact histories are data recorded when employees communicate with customers. By analyzing contact histories, we can determine business confidence levels in the local region and analyze inter-industry relations using industrial data that is attached to the contact history. However, it is often difficult for bankers to create analysis programs. Therefore, we propose a banker-friendly inter-industry relations analysis framework. In this study, we generated regional business confidence indices and used them to analyze inter-industry relations.

1 Introduction

In economics and finance, various data are used to forecast and analyze future market trends. The analysis methodology for financial forecasting is generally divided into technical analysis and fundamental analysis, depending on the type of data used. The former analyzes historical transaction prices and volume, while the latter involves a wider range of information, including forecasts of a company's business performance in addition to the data used in technical analysis. Although investors and analysts use the above-mentioned methodologies in conjunction with one another, they focus more on numerical data than on textual information, as the former is simpler to handle. However, the latter can also be useful for market analysis. For example, economic analysis reports written by financial experts provide rich data, while newspaper articles report important information concerning past events and their impact. In addition, comments on social networking sites reflect people's impressions of the economy. The demand for using textual information to forecast future trends is increasing, and a number of studies using machine learning have been conducted.

Local banks are also an important source of financial text data. For example, banks generate demand histories, financial summaries, and contact histories. In this study, we focus on contact histories, which document customers' authentic views concerning businesses and local economic conditions.

These data can provide important information about the industry sector. In this study, we aim to analyze relations between industry sectors in a region by using contact histories to create business confidence indices conveying industry sector information. These indices can serve as valuable tools for visualizing local economic conditions and evaluating local industries, which can be useful for the revitalization of local communities. Furthermore, business confidence indices are useful as reference indices for investment or policy decisions, as they capture economic trends.

In this paper, we propose a novel framework for visualizing local economic and local inter-industry relations using contact histories. Local business confidence indices currently exist¹; however, they often lack immediacy, as they reflect conditions from several months prior to their publication. With our proposed framework, however, it is possible to use text data to create an immediacy index to analyze local inter-industry relations. Furthermore, our framework is the first to generate business confidence indices using the contact histories of local banks.

The contribution of this study is to provide a framework for generating business confidence indices and analyzing inter-industry relations based on text mining techniques. This study makes the following contributions: 1) a means of generating business confidence indices that enables the frequent presentation of data while revealing correlations with existing indicators; 2) the visualization of business conditions based on generated indices in a manner that permits the analysis of inter-industry relations; and 3) a clarification of the effectiveness of the data owned by local banks, which have not been utilized up to now. This study is the first to analyze inter-industry relations using text data, and the usefulness of this approach is demonstrated by applying it to real data.

1.1 Related work

Bollen et al. have demonstrated that Twitter moods are useful for forecasting the Dow Jones Industrial Average [Bollen *et al.*, 2011]. The researchers used a self-organizing fuzzy neural network for forecasting with which they were able to predict rise and fall with an accuracy of over 80%. Schumaker et al. proposed a machine learning approach for predicting stock prices using financial news article analysis

¹http://www.okigin-ei.co.jp/report_DL.html

[Schumaker and Chen, 2009]. Their method predicted indicators and stock prices; however, it did not analyze inter-industry relations.

With regard to financial text mining, Sakai et al. proposed a method for extracting causal information from Japanese financial articles concerning business performance [Sakai and Masuyama, 2007]. Their method used clues for extracting causal information, and it was able to automatically gather clues using the bootstrapping method. Sakaji et al. proposed a method for automatically extracting basis expressions indicating economic trends from newspaper articles using a statistical approach [Sakaji *et al.*, 2008]. In addition, Koppel et al. proposed a method for classifying a company’s news stories on the basis of their apparent impact on the company’s stock performance [Koppel and Shtrimerberg, 2006]. Ito et al. proposed a neural network model for visualizing online financial textual data [Ito *et al.*, 2018]; this model determined the sentiment of words and their categories. Lastly, Milea et al. predicted the MSCI euro index (upwards, downwards, or constant) based on fuzzy grammar fragments extracted from a report published by the European Central Bank [Milea *et al.*, 2010].

The above-mentioned studies all extract information for investors or predict stock prices using information extracted from text data. In the present study, however, our objective is to analyze inter-industry relations using the contact histories of a local bank.

2 Proposed framework

In this section, we describe our proposed framework, which generates local business confidence indices and analyzes local inter-industry relations using the contact histories of local banks. Our framework uses a learned bidirectional long short-term memory (BiLSTM) model [Graves and Schmidhuber, 2005] for generating business confidence indices from banks’ contact histories. It then analyzes inter-industry relations using the generated business confidence indices.

The procedure of our proposed framework is as follows.

Step 1: As raw input data, text related to finance and the economy are selected (e.g., newspaper articles, economic trend surveys) that reflects local economic conditions. Then, a word embedding model is created from the input. Next, our method assigns monthly sentiment scores to the inputs using the created word embedding model and learned BiLSTM. Finally, business confidence indices are generated by summing each month’s sentiment score.

Step 2: Our method analyzes inter-industry relations using Granger causality analysis, impulse response analysis, and forecast error variance decomposition (FEVD) of the generated business confidence indices.

In Step 1, we generated word embeddings with 200 dimensions and default settings using *gensim*. In addition, we assumed that each raw input data entry corresponded to an industry sector. In this step, by changing the scale, our framework was also able to generate weekly indices. An overview

of our framework is presented in Figure 1. In our framework, by inputting text data, users can obtain four types of generated data: business confidence indices, graphs of inter-industry relations, graphs of inter-industry impacts, and graphs of FEVD.

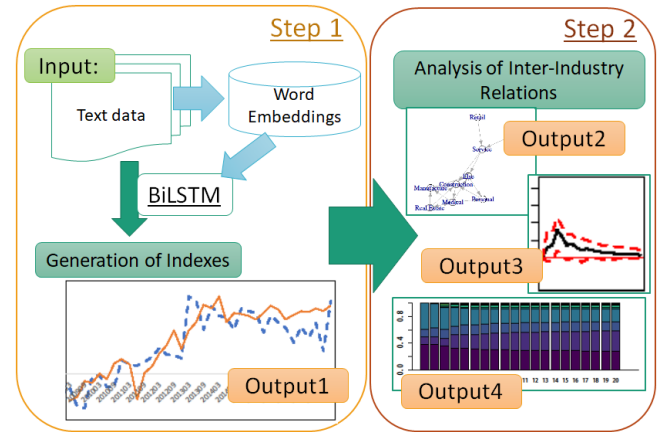


Figure 1: Proposed framework.

Although other machine learning methods exist, our framework uses BiLSTM, as it is a model in which long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] is bidirectional. A detailed explanation of the machine learning method selection is provided in Section 2.1.

2.1 Machine learning method selection

To select a machine learning method, we experimented with a classification test using the Economy Watchers Survey², which provides a timely and accurate overview of regional economic trends. The Economy Watchers Survey is scored on a 1–5 point Likert scale according to the level of business confidence, and each response is accompanied by comments. Using the survey responses, we constructed a machine learning method that predicts points using comments as input. To the best of our knowledge, the Economic Watcher Survey was the only available source of tagged text data on economics with a sufficient amount of information.

In this study, we used logistic regression (LR), random forest (RF), multi-layer perceptron (MLP), LSTM, and BiLSTM for classification testing. We used 20,000, 5,000, and 5,000 pairs of points and comments as training data, validation data, and test data, respectively. The pairs of points and comments were randomly extracted from the Economy Watchers Survey from 2010 to 2017. The results are presented in Table 1.

Table 1 reveals that BiLSTM is the optimal machine learning method for classifying the Economy Watchers Survey; as a result, it was adopted in our framework.

2.2 Bidirectional LSTM

Figure 2 illustrates our BiLSTM model.

As input, we used word embeddings of content words (nouns, verbs, and adjectives) selected using morphological

²<https://www5.cao.go.jp/keizai3/watcher-e/index-e.html>

Table 1: Classification test results

	Accuracy
LR	0.612
RF	0.525
MLP	0.617
LSTM	0.636
BiLSTM	0.642

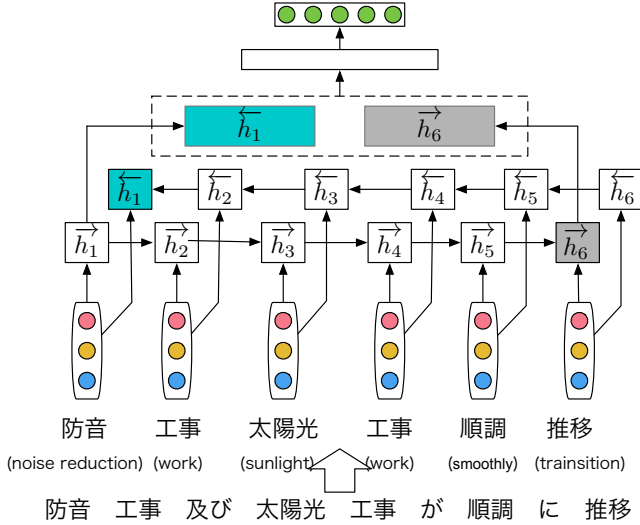


Figure 2: BiLSTM model.

analysis. Here, we define LSTM processing from the beginning of a sentence as \overrightarrow{LSTM} and from the end of the sentence as \overleftarrow{LSTM} . For each input, our method obtains $\{\overrightarrow{h_i}\}_i^n$ and $\{\overleftarrow{h_i}\}_i^n$ through $LSTM(\overrightarrow{LSTM}, \overleftarrow{LSTM})$.

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(e_i), \overleftarrow{h_i} = \overleftarrow{LSTM}(e_i) \quad (1)$$

Here, n is the number of input words, and e_i is the word embedding entered i th words.

Then, $\overleftarrow{h_1}$ and $\overrightarrow{h_n}$ are concatenated and entered into the output layer as follows:

$$s = [\overleftarrow{h_1}; \overrightarrow{h_n}] \quad (2)$$

$$t = \tanh(W_s \cdot s + b_s) \quad (3)$$

$$Y = W_t \cdot t + b_t \quad (4)$$

Here, $h_1 \in \mathbb{R}^m$, $h_n \in \mathbb{R}^m$, $s \in \mathbb{R}^{2m}$, and $t \in \mathbb{R}^l$.

Here, W_s and W_t are weighted matrices, b_s and b_t are bias vectors, m is the number of units in the hidden layer, l is the number of units in the middle layer, and Y is an output layer comprising $Y = (y_1, y_2, y_3, y_4, y_5)$. In this study, the output layer is activated by formula 5.

$$\beta_i = \log \left(\frac{\exp(y_i)}{\sum_j \exp(y_j)} \right) \quad (5)$$

Here, β_i is the activated output layer, and y_i is the output layer passed to the activation function. Finally, our method selects y_n as the maximum value from the output layer as output.

2.3 Granger causality analysis

We determined the causal relations between industry types using a Granger causality test [Granger, 1969]. Specifically, the null hypothesis is that there is no Granger causality from Industry A to B, and whether to accept or reject this hypothesis is determined by comparing its p value with the significance level. In this study, we used partial Granger causal analysis [Guo *et al.*, 2008], which extends Granger causality to multivariate data. Partial Granger causality analysis reduces the influence of exogenous factors and latent variables, and is useful in multivariate data analysis.

2.4 Impulse response analysis

The Granger causality test is a method for determining the causality between time series data; however, it cannot measure the strength of a relationship. The impulse response function is used as a means to quantitatively capture a relationship. By analyzing the degree to which changes in one variable contribute to changes in other variables, it is possible to perform a quantitative analysis. In the proposed framework, the orthogonalized impulse response function [Sims, 1980] is used and analyzed to quantitatively evaluate how changes in one industry affect other industries based on the business confidence index for each industry category.

2.5 Forecast error variance decomposition

FEVD is a method used to quantitatively analyze relationships between variables, and it is often used for the same purpose as impulse response analysis. Impulse response analysis is a method used to measure the influence of the fluctuation of a variable on other variables. In contrast, FEVD is used to analyze the business cycle of industries. It analyzes the relationships between variables by clarifying the degree to which each variable contributes to the forecast error.

In this study, we assumed that economic flow can be understood by quantitatively evaluating the contribution of each type of industry to changes in the business condition index of other types of industries.

3 Application

In this study, we focus on Okinawa Prefecture in Japan as an example of utilizing financial or economic text information and analyzing inter-industry relations within a region.

3.1 Data

The text data used in this study is as follows.

News texts published by Ryukyu Shimpo³, a local newspaper in Okinawa Prefecture in Japan. In this study, newspaper articles from Ryukyu Shimpo from January 2014 to December 2017 were used as data. Due to its geographical specificity and historical background, Okinawa Prefecture has the lowest penetration of national newspapers in Japan. The primary reason for targeting this local newspaper for analysis is its large regional influence.

³<https://ryukyushimpo.jp>

Contact history owned by Okinawa Bank, a local bank in Okinawa Prefecture in Japan. Contact history is the text data recorded when a bank employee communicates with customers. Thus, the unit of a contact history is a transaction. Contact histories are categorized into industry sectors by local banks; examples of contact histories are provided in Table 2. Economic circulation is supported by the flow of money, most of which occurs via financial institutions. In other words, data owned by a bank representing a local area is assumed to reflect the local economic entity. A contact history records not only actual transactions, but also customer backgrounds and their views on economic and business conditions. Therefore, it comprises useful data to evaluate the levels of business confidence. In this study, we used approximately 8 million contact histories from April 2011 to July 2017.

Table 2: Examples of contact histories

他行住宅ローン借換推進で訪問するが不在。名刺、チラシ投函する。(I visited for a mortgage loan promotion, but the client was not present. So, I put cards and flyers.)
現況確認 台風の影響で前期は売上減少 (Confirming the present condition, sales decreased in the previous year owing to typhoons.)
キャッシュカード暗証番号変更で来店。定期商品や保険を少しご案内しました。老後のことが不安との事なので時間があるときに検討する。(Customer visited us to change the cash card security code. We introduced a few regular products and insurance. The customer said, "I will consider these products when I have time because I am anxious about old age.")

3.2 Generation of business confidence index

News text data

Using text data from newspaper articles, in this experiment, the generated business confidence index was evaluated as two viewpoints. To evaluate the effectiveness of the generated business confidence index by unit change of text data, business confidence was supplied for various units of text data, such as news articles, as follows: (a) sentences, (b) paragraphs, (c) economic articles, and (d) all articles. Therefore, using the method proposed in [Sakaji *et al.*, 2008], text data was divided into roughly four types of units, and a business confidence index was generated using each unit.

Contact histories

he business confidence index was derived from the contact history on the basis of the sentiment classification model learned from the Economic Watcher Survey data. As described in Section 2.1, by performing sentiment classification using the Economic Watcher Survey, an optimum sentiment classification model was produced, and the adopted model calculated the sentiment value in the contact history. Five sentiment values were provided: good, slightly good, neutral, somewhat bad, and bad. The average sentiment value of a

certain period was then calculated as the business confidence index. An analysis of the inter-industry relationship was then performed on the basis of the generated business confidence index.

4 Result and discussion

4.1 Evaluation of generated business confidence index

The generated business confidence index was evaluated through a comparison with an existing index, such as the Okigin cooperation trend survey in this study. Thus, the correlation coefficient r calculated in (6) was adopted as the evaluation criteria.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (6)$$

Here, x is the generated index value, and y is the existing index value. Additionally, \bar{x} is the average value of the generated index, while \bar{y} is the average value of the existing index.

Table 3 presents the correlations between the generated business confidence indices and the existing index. Additionally, Figures 3 and 4 present the generated indices.

Table 3: Correlations between generated business confidence indices and existing index

	Correlation
(a)Per sentence	0.301
(b)Per paragraph	0.275
(c)Per article (economic)	0.300
(d)Per article (all)	0.175
Contact histories	0.856

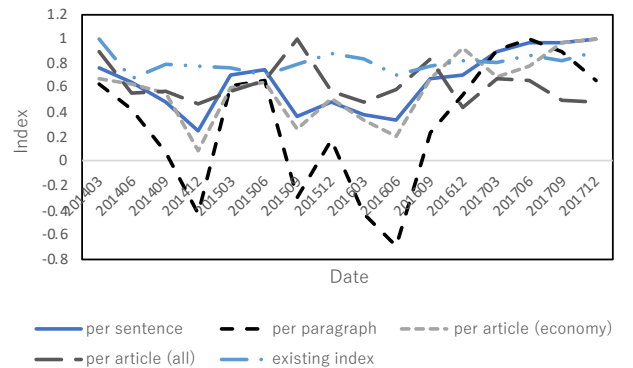


Figure 3: Generated index based on news text data.

Table 3 indicates that the index generated from the contact histories outperforms the generated indices from the news text. The reason for this is most likely the high affinity of the contact histories. Contact histories occasionally include customers' levels of business confidence; therefore, a business confidence index can be generated effectively using contact histories.

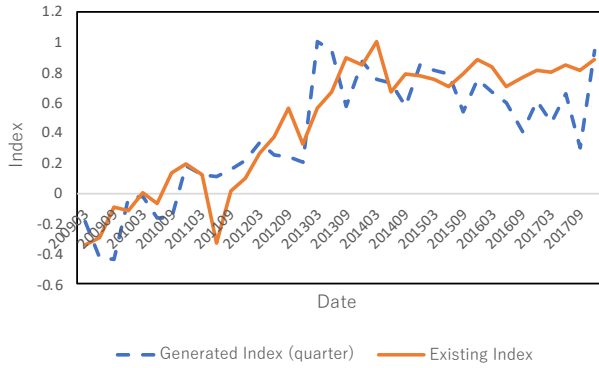


Figure 4: Generated index based on the contact histories.

4.2 Analysis of inter-industry relations

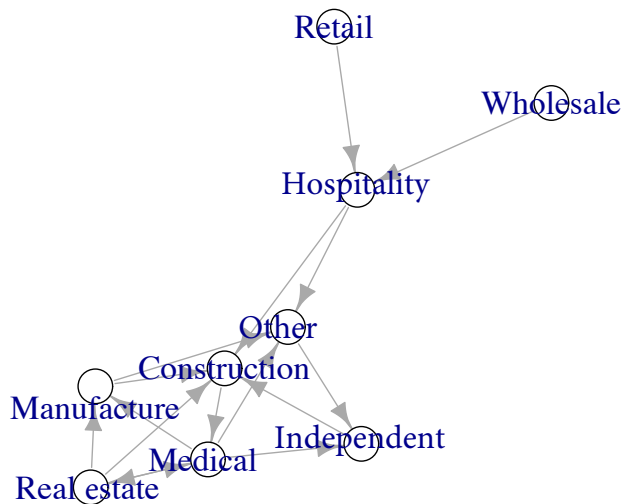


Figure 5: Visualization of partial Granger causality analysis among industries

function of the hospitality industry sector on the other industries, while dashed red lines represent the two-sided 95% confidence interval. In Figure 6, the horizontal and vertical axes represent the monthly and standard deviation, respectively. This figure indicates that the hospitality industry has affected business confidence in other industries for a long period of time. For example, at the 20th term in Figure 6, the manufacturing and wholesale industries have a high score (0.004).

Figure 7 presents the FEVD results for business confidence by industry sector. In this Figure 7, the vertical axis indicates the contribution of the influence by other industries, while the horizontal axis indicates monthly. In Figure 7, it can be seen that the retail, wholesale, and hospitality industries have

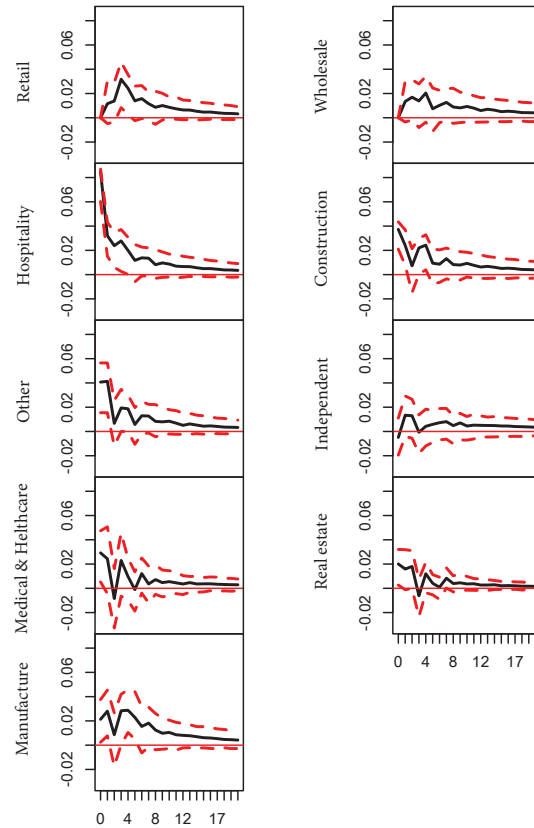


Figure 6: Impulse responses (hospitality industry).

a large impact on other industries.

In this study, we validated these results by comparing the production ripple effects between industries. As a result, the production ripple effect to be compared is related to the power of dispersion and the production inducement coefficient in Okinawa Prefecture. The power of dispersion is an index that quantifies the production ripple effect on an entire sector when one unit of final demand occurs in one sector. In contrast, the production inducement coefficient is an index that quantifies the production ripple effect on each sector when one unit of final demand occurs in all sectors.

In Figure 5, the production inducement coefficients of industries in Okinawa Prefecture are listed in descending order of magnitude as follows: commercial (retail and wholesale businesses) > construction industry > hospitality > manufacturing > medical care, health care, social security, nursing care > real estate. We analyzed this order in conjunction with the analysis results of the changes in business confidence. The top three industries (commercial, construction, and hospitality) correspond to the industries located upstream in Figure 5, and they appear as major fluctuation factors for other industries in the impulse response analysis. This was also confirmed by the FEVD results. A large power of dispersion has the effect of promoting production for other industries. The improvement in business confidence in these industries stimulates production and enhances business con-

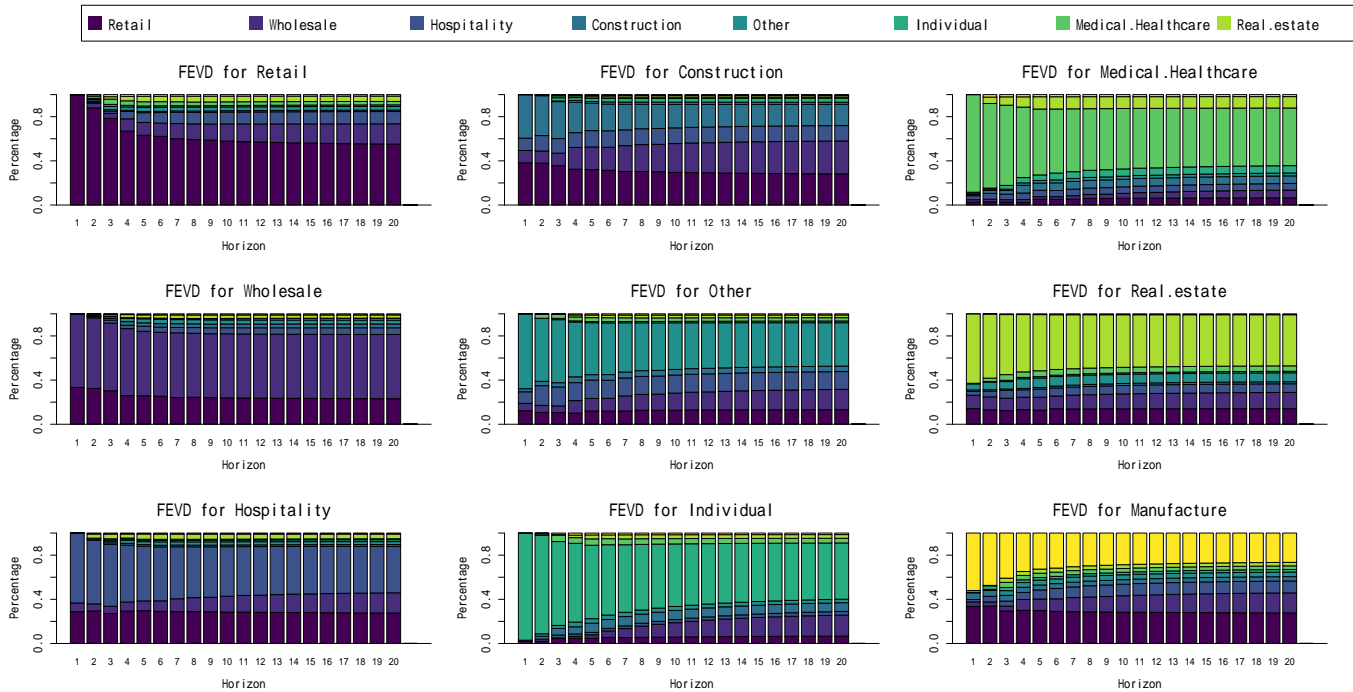


Figure 7: Results of forecast error variance decomposition (FEVD).

confidence in other industries. This propagation is reasonable, as industries that cause changes in business confidence in other industries correspond to industries with a high production ripple effect.

Next, we analyzed relationships using the production inducement coefficient. Six industries are listed in descending order of the production inducement coefficient as follows: hospitality > manufacturing > commercial (retail and wholesale) > real estate > construction industry > medical care, health care, social security, nursing care. The results of the partial Granger causal test demonstrate that the top two industries are influenced by a large number of industries. Figure 5 reveals that manufacturing is significantly affected by three industries, while hospitality is significantly affected by two industries. When the analysis is correlated with the results of FEVD presented in Figure 7, it is difficult to explain any unpredictable changes in the hospitality and manufacturing industries owing to the fact that both the sensitivity coefficient and contribution from other industries are high. In other words, the hospitality and manufacturing industries are strongly affected by production activities in other industries and are highly sensitive to economic fluctuations in those industries.

5 Conclusion

In this study, we proposed a framework for generating a business confidence index and analyzing the inter-industry structure of a local area, using text data representing the characteristics of the economy of the local area. We demonstrated that the business confidence index generated using the con-

tact history owned by a local bank can reproduce the existing index with high accuracy. In addition, unlike the existing index, the business confidence index in a local area can be obtained more frequently than other text sources. Furthermore, because category classification was performed for the contact history owned by the local bank in question, it was possible to obtain the business confidence for each industry category.

In this study, we used Granger causal analysis, impulse response function analysis, and variance decomposition methods to analyze the causality of different time series data from the obtained business confidence index for each industry category. The results revealed the changes in business confidence among industries, the effect of the business confidence index on each industry category, and the contribution of each industry category to other industries.

In future work, we plan to use other forms of data to generate business confidence indices. In addition, we intend to adapt our framework to data from another local bank as a means to better evaluate our framework and expand our understanding of business confidence indices.

References

- [Bollen *et al.*, 2011] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.
- [Granger, 1969] C.W.J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37:424–438, 1969.
- [Graves and Schmidhuber, 2005] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with

- bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.
- [Guo *et al.*, 2008] S. Guo, A.K. Seth, K.M. Kendrick, C. Zhou, and J. Feng. Partial granger causality—eliminating exogenous inputs and latent variables. *Journal of Neuroscience Methods*, 172(1):79–93, 2008.
- [Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [Ito *et al.*, 2018] Tomoki Ito, Hiroki Sakaji, Kiyoshi Izumi, Kota Tsubouchi, and Tatsuo Yamashita. Ginn: gradient interpretable neural networks for visualizing financial texts. *International Journal of Data Science and Analytics*, Dec 2018.
- [Koppel and Shtrimberg, 2006] Moshe Koppel and Itai Shtrimberg. *Good News or Bad News? Let the Market Decide*, pages 297–301. Springer Netherlands, Dordrecht, 2006.
- [Milea *et al.*, 2010] Viorel Milea, Nurfadhlin Mohd Sharef, Rui Jorge Almeida, Uzay Kaymak, and Flavius Frasinca. Prediction of the msci euro index based on fuzzy grammar fragments extracted from european central bank statements. In *2010 International Conference of Soft Computing and Pattern Recognition*, pages 231–236, Dec 2010.
- [Sakai and Masuyama, 2007] Hiroyuki Sakai and Shigeru Masuyama. Extraction of cause information from newspaper articles concerning business performance. In *Proc. of the 4th IFIP Conference on Artificial Intelligence Applications & Innovations*, pages 205–212, 2007.
- [Sakaji *et al.*, 2008] Hiroki Sakaji, Hiroyuki Sakai, and Shigeru Masuyama. Automatic extraction of basis expressions that indicate economic trends. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 977–984, 2008.
- [Schumaker and Chen, 2009] Robert P. Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Trans. Inf. Syst.*, 27(2):12:1–12:19, March 2009.
- [Sims, 1980] Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pages 1–48, 1980.

Learning to Learn Sales Prediction with Social Media Sentiment

Zhaojiang Lin¹, Andrea Madotto¹, Genta Indra Winata¹, Zihan Liu¹, Yan Xu¹,
Cong Gao¹ and Pascale Fung^{1,2}

¹Center for Artificial Intelligence Research (CAiRE)

Department of Electronic and Computer Engineering

The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

²EMOS Technologies Inc

{zlinao, amadotto, giwinata, zliucr, yxucb, eemiranda}@connect.ust.hk,
pascale@ece.ust.hk

Abstract

Social media sentiment has shown to be a useful resource for product sales forecast. However, research on modeling the correlation between sentiment index and sales is often limited by the scarceness of quarterly sales data. In this paper, we propose to learn how to learn sentiment-sales correlation from different source products and transfer to sales prediction of another, target product. We evaluated our approach on sales data of seven different smartphones and showed that the knowledge transfer from six source products significantly reduced the sales prediction error for the target product, in a 7-fold cross-validation experiment.

1 Introduction

The sales forecast is crucial in the financial domain since it indicates the future trend of a product and thus, it allows investors to make better decisions. During the years, time-series models have been widely applied in sales prediction using historical sales data. However, they are often unreliable since historical sales ignore the importance of customers opinions (e.g., Social Media, News), which are critical for sales prediction [Ahn and Spangler, 2014]. On the other hand, user-generated content in social media acts as word of mouth contains a large number of customer opinions. Sentiment analysis of social media provides a good summary of customers' feedback and allow companies to have a better intuition of how the market reacts to their products.

Several existing work use sentiment features to predict product sales, for instance for movie sales [Duan *et al.*, 2008; Gaikar and Marakarkandy, 2015; Ahn and Spangler, 2014; Marshall *et al.*, 2013; Asur and Huberman, 2010], e-commerce products [Davis and Khazanchi, 2008; Tuarob and Tucker, 2013] and car sales [Wijnhoven and Plant, 2017; Geva *et al.*, 2017; Barreira *et al.*, 2013]. These works show positive correlations between sentiments features and sales, and thus, the sentiment is a useful indicator to predict the outcome of future sales. However, most of them focus on correlation studies between features, but they have not explored the possibility to transfer information from different brands.

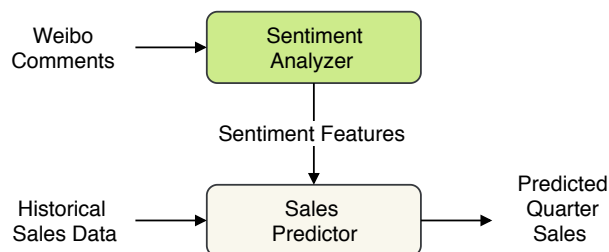


Figure 1: Overall architecture. In a big picture, it consists of sentiment analyzer to extract sentiment features from Weibo comments and they are fed into the sales prediction model along with historical sales data.

Moreover, the preferred models for the sales prediction task are usually linear (e.g., BIC, ARIMA) due to the particularly small datasets. Indeed, a well-known problem for deep learning models, and in general non-linear models, is that they require a large amount of data to work properly.

Differently from the previous work, in this paper, we study the sales of seven smartphones in China's market such as Samsung, Gionee, Huawei, Oppo, Vivo, Meizu, and iPhone. We show the importance of sentiment features by incorporating sentiment information – extracted from the biggest Chinese social media platform Weibo – for improving sales prediction. To extract reliable sentiment index from Weibo, we build an accurate sentiment analyzer by applying the state-of-the-art pre-trained model BERT: Bidirectional Encoder Representation from Transformer [Devlin *et al.*, 2018]. Moreover, we report the sales prediction results of several statistical models and show the usefulness of sentiment features. Most importantly, we propose a viable way to alleviate the scarceness of sales data by using meta-learning. This technique allows a non-parametric model such as neural networks to leverage historical sales of other brands, and use them as the prior knowledge. The intuition of applying meta-learning is that it optimizes the model for fast adaptability, allowing it to adapt to new prediction tasks.

The main contributions of this paper are: 1) Collecting and pre-processing a large scale dataset of user comments of seven different smartphone companies from a popular Chinese social media platform Weibo, and providing human la-

Brands	# Comments
Samsung	288,081
Gionee	302,866
Huawei	524,468
Oppo	633,406
Vivo	670,408
Meizu	945,312
iPhone	994,155

Table 1: The number of Weibo comments for smartphone brands.

beled sentiment annotations of 25K comments; 2) Training a state-of-the-art sentiment classifier to produce reliable sentiment features for the sales prediction; 3) Reporting consistent improvements in the sales prediction by using the extracted sentiments features, which confirms existing related previous works; 4) Proposing a deep learning-based solution that can compete, and improve, with very strong statistical models. By using meta-learning, our model is able to leverage other brands sales history for making a more accurate sales prediction. To the best of our knowledge, we are the first to report positive results in this setting.

In the following sections, we introduce 1) Corpus collection and annotation, and the historical sales dataset used in our experiments; 2) Sentiment analyzer and sales prediction models; 3) Experiments and results; 4) Related work; and 5) Conclusion.

2 Dataset Collection

2.1 Weibo Sentiment Dataset

We crawl around 5 million Weibo comments for seven different smartphones: Samsung, Gionee, Huawei, Oppo, Vivo, Meizu, and iPhone from their company official accounts from 2013 to 2018. In the data cleaning process, we remove all emojis, user mentions such as “@user”, hashtags, and hyperlinks using regular expressions. Then, we group them by quarter, a period of four months. The statistics of the dataset for each brand is showed in Table 1.

We randomly sample 25,000 Weibo comments and manually annotated them with *Positive*, *Negative*, and *Neutral* labels via crowd-sourcing. The agreement is taken by majority vote. The annotation result shows that the percentage of *Positive*, *Negative*, and *Neutral* labels are 20%, 16%, and 64% respectively. We further take around 5,000 comments as our test set.

2.2 Smartphone Sales Dataset

We collect quarterly China sales data of seven smartphones: Samsung, Gionee, Huawei, Oppo, Vivo, Meizu, and iPhone from the first quarter of 2013 to the third quarter of 2018 released by IDC¹. In each brand, we reserve the last five quarters for testing, and we use the rest for training our models.

¹<https://www.idc.com/>

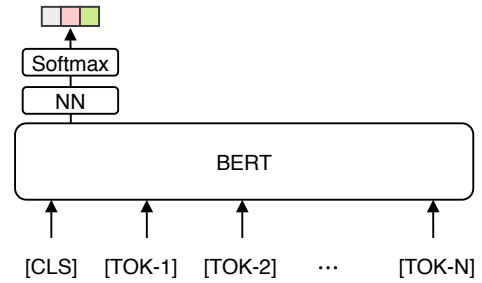


Figure 2: Sentiment Analyzer. The model accepts user comment tokens and generate a probability distribution over three classes. In the figure, green states for positive, red for negative, and gray for neutral.

3 Methodology

3.1 Sentiment Analysis

Building a reliable sentiment classifier is crucial to the final sales prediction. To alleviate the dependence of the human effort and build a robust sentiment classifier, we apply current state of the art pre-trained language model BERT: Bidirectional Encoder Representations from Transformers [Devlin *et al.*, 2018] to our task. It is a multi-layer bidirectional Transformer encoder pre-trained by using “masked language model” objective. In our proposed model, we adapt $BERT_{BASE}$ ² [Devlin *et al.*, 2018] to generate the semantic representation of each comment to improve the sentiment prediction task. Following the same fine-tuning procedure of [Devlin *et al.*, 2018], a special token $[CLS]$ is added at the beginning of every input to obtain the fixed-dimensional representation of input sequence. As shown in Figure 1, we stack another linear layer with Softmax function on top of $BERT$ to compute the probabilities of three sentiment classes. We keep all parameters trainable, and they are fine-tuned with the sentiment training data. Our sentiment analyzer achieves around 80% accuracy in the final test set.

Sentiment Features To incorporate sentiment information into the sales predictor, we quantify the sentiment score of each brand in the quarter. We calculate the score x_t by the following [Lassen *et al.*, 2014]:

$$x_t = \frac{p_t}{p_t + n_t} \quad (1)$$

where p_t is the number of comments with positive sentiment in the quarter t , and n_t is the number of comments with negative sentiment in the quarter t . The score is normalized to 0-1 range.

3.2 Sales Prediction

Let us define a vector $S = [s_0, \dots, s_t, \dots, s_N]$ as the sales at each quarter and vector $X = [x_0, \dots, x_t, \dots, x_N]$ as sentiment features at each quarter, where N is the total number of quarters, s_t is the sales value at quarter t , and x_t is the sentiment of comments posted in one month time before in each quarter. For example, in the second quarter of a year from

²We used a PyTorch implementation from <https://github.com/huggingface/pytorch-pretrained-BERT>

April to June, we use sentiment of the comments posted from March to May. The task of our model is to predict sales s_t by taking in input the sales history $S_{0:t-1} = [s_0, \dots, s_{t-1}]$ and current sentiment value x_t . In this section, we introduce two different approaches: (1) a statistical-based model, Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors (*SARIMAX*) (2) a gradient-based model, Multi-layer Perceptron (*MLP*). We also describe meta-learning procedure in our sales prediction task.

SARIMAX

The model is an extension of *SARIMA* model with external variables. We denote the model by $SARIMAX(p, d, q)(P, D, Q)_S(X)$, where p, d, q are orders of autoregressive, difference, and moving average and P, D, Q are orders of seasonal autoregressive, difference, and moving average. X is the external variable and S is the seasonal period (e.g., quarter). The quarterly sales series $S_{0:t}$ is computed given sentiment features x_t as follows:

$$S_{0:t} = \frac{\theta_q(B)\Theta_Q(B^S)}{\phi_p(B)\Phi_P(B^S)(1-B)^d(1-B^S)^D} \varepsilon_t + y_t, \quad (2)$$

$$y_t = w_0 + w_1 x_t \quad (3)$$

where w_0 and w_1 are regression coefficients, B and B^S are delay operators, $\phi_p(B)$ is a non-seasonal autoregressive operator with p -order, $\Phi_P(B^S)$ is a seasonal autoregressive operator with P -order, $\theta_q(B)$ is a non-seasonal moving average operator with q -order, $\Theta_Q(B^S)$ is a seasonal moving average operator with Q -order, and ε_t is a residual error.

MLP

MLP consists of multiple linear layers followed by a nonlinear activation function. Unlike autoregressive model *SARIMAX*, *MLP* requires a fixed-dimensional feature input. Therefore we take the sentiment feature along with last four quarters historical sales number as our input feature:

$$s_t = f(S_{t-5:t-1}, x_t; \theta) \quad (4)$$

where f is *MLP* model parameterized by θ .

Meta-Learning

In this work, we apply Model-Agnostic Meta-Learning (MAML) [Finn *et al.*, 2017] to sales prediction task. The goal of MAML in our task is to find initial parameters θ_0 of sales predictor model f_θ (*MLP* in our case) such that the model can make an accurate prediction on a new product after training on few historical sales samples.

In our meta-learning scenario, every product is considered as a different task. As we showed in the Figure 3, datasets \mathcal{D}_i are constructed separately for each task. We take one product out as meta-test set $\mathcal{D}_{meta-test}$, other datasets as meta-training set $\mathcal{D}_{meta-train}$. In meta-training setting of [Finn *et al.*, 2017], for each dataset \mathcal{D}_i , they random sample some data points $\mathcal{D}_{i,train}$ for inner training and sample some other data points $\mathcal{D}_{i,dev}$ for meta-update. Instead, we always fix the split $\mathcal{D}_{i,train}$ and $\mathcal{D}_{i,dev}$, because we are only interested in forecasting sales given historical sales. During the meta training,

Algorithm 1 MAML for sales prediction task

Require: $\mathcal{D}_{meta-train}$

Require: α, β learning rate

- 1: Randomly initialize θ
- 2: **while** not done **do**
- 3: Sample batch of products $\mathcal{D}_i \sim \mathcal{D}_{meta-train}$
- 4: **for all** \mathcal{D}_i **do**
- 5: $(\mathcal{D}_{i,train}, \mathcal{D}_{i,dev}) \leftarrow \mathcal{D}_i$
- 6: Evaluate $\nabla_{\theta} \mathcal{L}_{\mathcal{D}_i}(f_{\theta})$ using $\mathcal{D}_{i,train}$ and $\mathcal{L}_{\mathcal{D}_i}$ in Equation (5)
- 7: Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{D}_i}(f_{\theta})$
- 8: **end for**
- 9: Update $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{D}_i \sim \mathcal{D}} \mathcal{L}_{\mathcal{D}_i}(f_{\theta'_i})$ using $\mathcal{D}_{i,dev}$ and $\mathcal{L}_{\mathcal{D}_i}$ in Equation (5)
- 10: **end while**

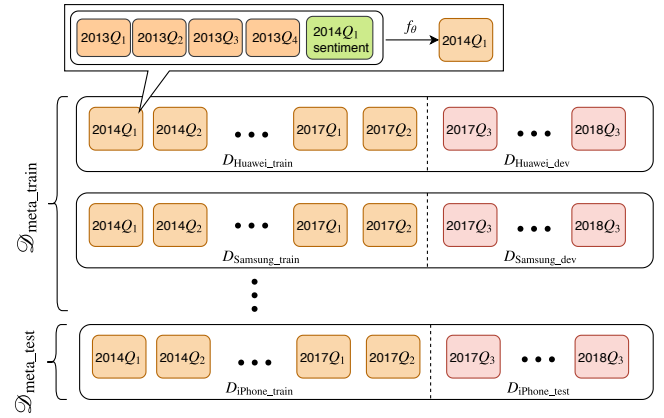


Figure 3: Example of meta-learning for sales prediction. The goal is to predict iPhone sales in next quarters. Meta-learning uses a series of historical data and sentiment from other smartphone brands to initialize the predictor model.

the model keeps simulating learning process that minimizes the prediction error by utilizing the historical training samples. The prediction error is measured by MSE (Mean Square Error) defined by equation (5). We describe the learning procedure in Algorithm 1. After meta-learning, we train our model on historical sales data $\mathcal{D}_{i,train}$ from meta-training set $\mathcal{D}_{meta-train}$, and finally evaluate our model on $\mathcal{D}_{i,dev}$ from $\mathcal{D}_{meta-test}$.

$$\mathcal{L}_{\mathcal{D}_i}(f_{\theta}) = \sum_{\mathbf{x}^{(j)}, \mathbf{y}^{(j)} \sim \mathcal{D}_i} \|f_{\theta}(\mathbf{x}^{(j)}) - \mathbf{y}^{(j)}\|_2^2 \quad (5)$$

4 Experimental and Results

4.1 Settings

In our experiments, we compare the sales prediction performance of our models with and without using sentiment information, with and without using the meta-learning method in our smartphone sales dataset. We also compare our model with two baselines: linear regression and SVR (Support Vector Regression). As mentioned in the dataset section we use

MSE	iPhone	Gionee	Huawei	Meizu	Oppo	Samsung	Vivo	Average
<i>Linear</i>	10.956	23.733	4.139	5.745	12.639	2.702	7.779	9.670
<i>Linear+Sentiment</i>	5.501	4.082	7.420	6.749	13.215	2.251	7.394	6.659
<i>SVR</i>	7.733	7.533	4.37	6.044	4.764	9.203	8.672	6.903
<i>SVR+Sentiment</i>	4.106	4.444	4.714	11.836	6.869	4.532	9.107	6.515
<i>SARIMAX</i>	0.588	10.241	6.331	2.783	8.875	0.876	11.552	5.892
<i>SARIMAX+Sentiment</i>	0.072	8.232	6.667	5.742	2.114	1.073	10.869	4.967
<i>MLP</i>	15.429	8.565	3.684	6.55	11.03	0.737	9.931	7.990
<i>MLP+Sentiment</i>	3.625	3.128	3.187	6.199	2.782	0.891	16.648	5.209
<i>MLP+Sentiment+Meta</i>	0.822	2.765	4.906	9.114	3.525	1.145	7.134	4.202

Table 2: Results in Mean Squared Error (MSE).

Product	p	d	q	P	D	Q	S
iPhone	0	1	0	1	1	0	4
Gionee	0	1	0	1	0	0	4
Huawei	0	1	0	1	1	0	4
Meizu	0	1	0	1	1	0	4
Oppo	0	1	0	1	0	0	4
Samsung	0	1	0	1	0	0	4
Vivo	0	1	0	1	0	0	4

Table 3: SARIMAX hyper-parameters.

the last five quarters for testing and the previous for training. Hence, the model’s performance to predict the next quarter sales is evaluated using Mean Squared Error (MSE) of the test set.

Hyper-parameters SARIMAX model is identified by following hyper-parameters: order of difference (d), the order of seasonal difference (D), non-seasonal autoregressive order (p), seasonal autoregressive order (P), non-seasonal moving average order (q), seasonal moving average order (Q). All of them are identified by Autocorrelations function (ACF) and partial autocorrelations function (PACF) as we showed in Table 3. For our gradient base model, we use two layer MLP with hidden size 5 and Rectified Linear Units (ReLU) as the activation function. For meta-learning, we use SGD optimizer with learning rate 0.01 for both inner and outer optimization. we run 9 iterations for each inner update, and 10 epochs of meta update.

4.2 Results

Table 2 shows the results for each model and each brand in the term of Mean Squared Error (MSE). Two results stand out: Sentiments Features consistently improves the MSE for all the models, and the MLP with sentiment features trained using Meta-Learning can improve the average MSE among different brands.

Sentiment Features The features help in all the evaluated models, this confirms the usefulness of such a feature in sales prediction. Indeed, this shows that Weibo comments hold essential information that can be used to predict future sales. However, from Table 2 we can notice that the only case where sentiment features hurt the performance is on Meizu data. One possible reason could be the price of Meizu is much lower than other brands; hence the sentiment might not affect

the sales of low price products that target a different market.

SARIMAX vs MLP Moreover, in Table 2 we can see that both SARIMAX and MLP using sentiment features have a very similar average MSE and they performs consistently better than SVR and Linear Regression. Especially, MLP works the best for Huawei and Samsung where instead for iPhone, Oppo and Meizu SARIMAX works the best.

Meta-learning The best MSE average is achieved by the meta-learned model, MLP+Sentiment+Meta in Table 2. This is due to the ability to transfer knowledge between different brands. Indeed, meta-learning is trained to find a set of parameters that are able to quickly adapt to a given task. In our instance, this means to learn a set of parameters that can quickly adapt to the sales behavior of a certain brand.

Moreover, in Figure 4 we plot the Gionee, Vivo, Samsung and iPhone sales traces and the prediction made by MLP by using with and without sentiment feature including meta-learning to describe our findings. For Vivo, Samsung and iPhone, we can note that by just using MLP the sales predictions are not aligned with the real sales. Instead by adding sentiment features we can achieve a very good fit in the two quarters, but a more substantial error when a trend inversion appears (i.e., 2017Q4 in iPhone). This is mostly solved by meta-learning training, in which the model achieves almost a perfect fit (0.822 MSE).

We can also notice that in some brands predictions are easier than the others. For instance, the iPhone has seasonal patterns where there are peaks between the third and fourth quarters in the last two years. In this case, our autoregressive model SARIMAX can capture this pattern better than MLP with meta-learning as we showed in Table 2. On the other hand, SARIMAX predicts very poorly on Gionee and Vivo which have less repeating sales patterns. Conversely to our meta-learning based model is more robust as it can accurately predict in sales trends with irregular changes.

5 Related work

5.1 Sales prediction with sentiment analysis

Sentiment and emotional analysis are important methods to quantify customers’ emotional engagement [Winata *et al.*, 2019]. The importance and effectiveness of using social media opinion, a.k.a. Word-of-Mouth, for Sales Prediction is a well known topic [Hennig-Thurau *et al.*, 2003; Hennig-Thurau *et al.*, 2004; Ceron and d’Adda, 2016; Liu, 2012;

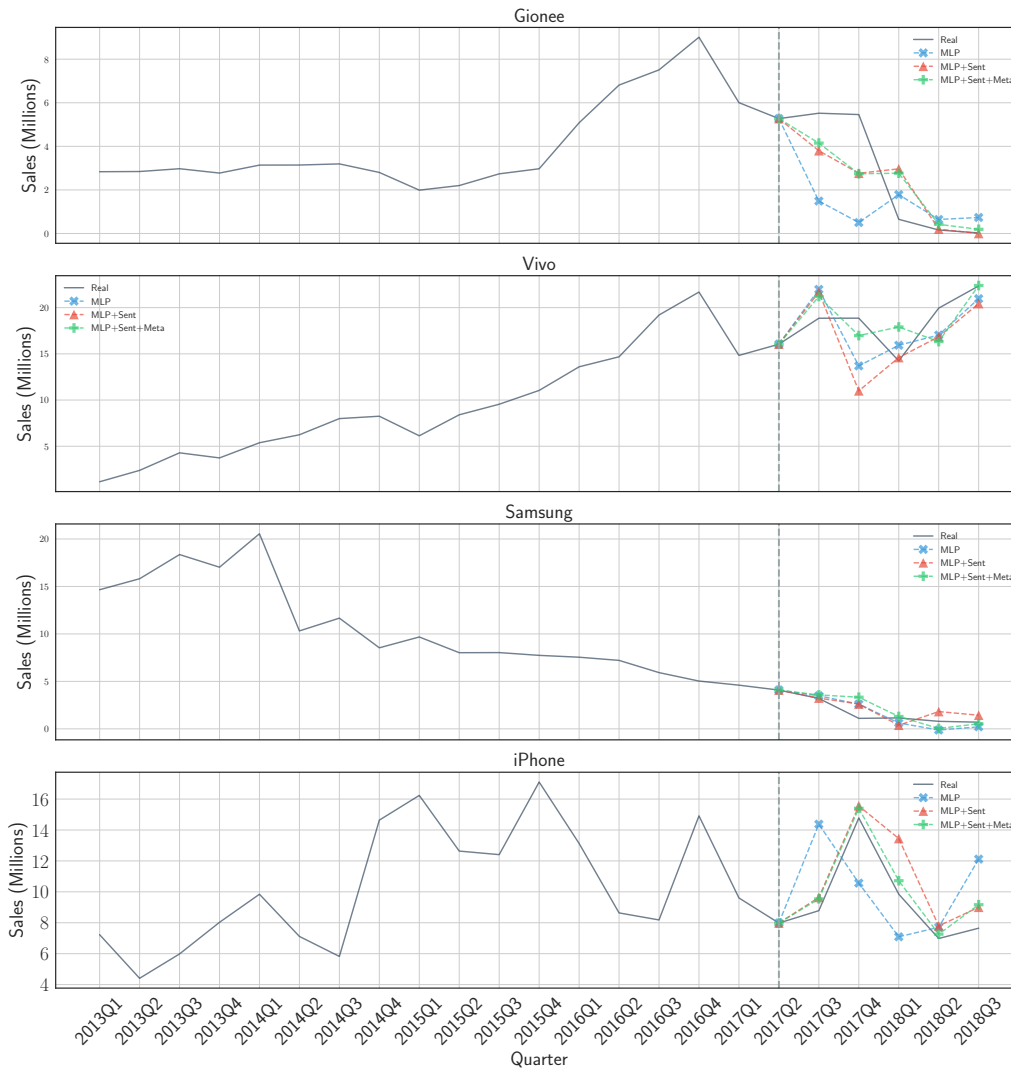


Figure 4: The sales prediction for Gionee, Vivo, Samsung and iPhone: Grey line represents the real sales, the blue line represents the prediction of MLP without sentiment information, the red line represents the prediction of MLP with sentiment information, and the green line represents the prediction of meta trained MLP with sentiment information.

Shi *et al.*, 2016; Asur and Huberman, 2010]. Among the years, using sentiment analysis as an additional features for sales forecasting has been widely used in different domains. For instance, it has been used for predicting: movies sales [Duan *et al.*, 2008; Gaikar and Marakarkandy, 2015; Ahn and Spangler, 2014; Marshall *et al.*, 2013; Asur and Huberman, 2010], e-commerce products [Davis and Khazanchi, 2008; Tuarob and Tucker, 2013], car sales [Wijnhoven and Plant, 2017; Geva *et al.*, 2017; Barreira *et al.*, 2013]. To the best of our knowledge we are the first to report positive correlation between sentiment feature and smartphones quarter sales.

5.2 Meta-learning

Meta-learning [Thrun and Pratt, 1998; Schmidhuber, 1987; Schmidhuber, 1992; Naik and Mammone, 1992; Bengio *et al.*, 1992] also known as Learning To Learn, is machine learn-

ing technique that tries to learn the algorithm itself. Recently, several meta-learning models has been proposed for solving few-shot image classification [Ravi and Larochelle, 2017; Vinyals *et al.*, 2016; Finn *et al.*, 2017; Mishra *et al.*, 2017; Santoro *et al.*, 2016], optimization [Andrychowicz *et al.*, 2016], dialogue system [Lin *et al.*, 2019] and reinforcement learning [Finn *et al.*, 2017]. In our setting, we are applying Meta-learning for learning a set of parameter that can adapt to certain products, and have good performance in sales prediction.

6 Conclusion

In this paper, we explore four different sales prediction models *SARIMAX*, *SVR*, *Linear Regression* and *MLP*. The results of our experiments show that sentiment information improves the performance of these models which confirms the effectiveness of the sentiment index. Moreover, the proposed

meta-learning method help models transfer the knowledge of sentiment-sales correlation from different products, further reduce the sales prediction error.

Acknowledgments

This work has been partially funded by ITF/319/16FP and MRP/055/18 of the Innovation Technology Commission, the Hong Kong SAR Government.

References

- [Ahn and Spangler, 2014] Hyung-II Ahn and W Scott Spangler. Sales prediction with social media analysis. In *2014 Annual SRII Global Conference*, pages 213–222. IEEE, 2014.
- [Andrychowicz *et al.*, 2016] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.
- [Asur and Huberman, 2010] Sitaram Asur and Bernardo A Huberman. Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 492–499. IEEE Computer Society, 2010.
- [Barreira *et al.*, 2013] Nuno Barreira, Pedro Godinho, and Paulo Melo. Nowcasting unemployment rate and new car sales in south-western europe with google trends. *NET-NOMICS: Economic Research and Electronic Networking*, 14(3):129–165, 2013.
- [Bengio *et al.*, 1992] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pages 6–8. Univ. of Texas, 1992.
- [Ceron and d’Adda, 2016] Andrea Ceron and Giovanna d’Adda. E-campaigning on twitter: The effectiveness of distributive promises and negative campaign in the 2013 italian election. *New media & society*, 18(9):1935–1955, 2016.
- [Davis and Khazanchi, 2008] Alanah Davis and Deepak Khazanchi. An empirical study of online word of mouth as a predictor for multi-product category e-commerce sales. *Electronic markets*, 18(2):130–141, 2008.
- [Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [Duan *et al.*, 2008] Wenjing Duan, Bin Gu, and Andrew B Whinston. The dynamics of online word-of-mouth and product sales—an empirical investigation of the movie industry. *Journal of retailing*, 84(2):233–242, 2008.
- [Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.
- [Gaikar and Marakarkandy, 2015] Dipak Gaikar and Bijith Marakarkandy. Product sales prediction based on sentiment analysis using twitter data. *Int. J. Comput. Sci. Inf. Technol.(IJCSIT)*, 6(3):2303–2313, 2015.
- [Geva *et al.*, 2017] Tomer Geva, Gal Oestreicher-Singer, Niv Efron, and Yair Shimshoni. Using forum and search data for sales prediction of high-involvement projects. *MIS Quarterly*, 41(1):65–82, 2017.
- [Hennig-Thurau *et al.*, 2003] Thorsten Hennig-Thurau, Gianfranco Walsh, and Gianfranco Walsh. Electronic word-of-mouth: Motives for and consequences of reading customer articulations on the internet. *International journal of electronic commerce*, 8(2):51–74, 2003.
- [Hennig-Thurau *et al.*, 2004] Thorsten Hennig-Thurau, Kevin P Gwinner, Gianfranco Walsh, and Dwayne D Gremler. Electronic word-of-mouth via consumer-opinion platforms: what motivates consumers to articulate themselves on the internet? *Journal of interactive marketing*, 18(1):38–52, 2004.
- [Lassen *et al.*, 2014] Niels Buus Lassen, Rene Madsen, and Ravi Vatrupu. Predicting iphone sales from iphone tweets. In *Enterprise Distributed Object Computing Conference (EDOC), 2014 IEEE 18th International*, pages 81–90. IEEE, 2014.
- [Lin *et al.*, 2019] Zhaojiang Lin, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Personalizing dialogue agents via meta-learning. *ArXiv*, abs/1905.10033, 2019.
- [Liu, 2012] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.
- [Marshall *et al.*, 2013] Pablo Marshall, Monika Dockendorff, and Soledad Ibáñez. A forecasting system for movie attendance. *Journal of Business Research*, 66(10):1800–1806, 2013.
- [Mishra *et al.*, 2017] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *ICLR*, 2017.
- [Naik and Mammone, 1992] Devang K Naik and RJ Mammone. Meta-neural networks that learn by learning. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 437–442. IEEE, 1992.
- [Ravi and Larochelle, 2017] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.
- [Santoro *et al.*, 2016] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

- [Schmidhuber, 1987] Jurgen Schmidhuber. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May 1987.
- [Schmidhuber, 1992] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.
- [Shi *et al.*, 2016] Xiaohui Shi, Feng Li, and Ali Ziaee Bigdeli. An examination of npd models in the context of business models. *Journal of Business Research*, 69(7):2541–2550, 2016.
- [Thrun and Pratt, 1998] Sebastian Thrun and Lorien Pratt, editors. *Learning to Learn*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.
- [Tuarob and Tucker, 2013] Suppawong Tuarob and Conrad S Tucker. Fad or here to stay: Predicting product market adoption and longevity using large scale, social media data. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V02BT02A012–V02BT02A012. Citeseer, 2013.
- [Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.
- [Wijnhoven and Plant, 2017] Alphonsus BJM Wijnhoven and Olivia Plant. Sentiment analysis and google trends data for predicting car sales. In *38th International Conference on Information Systems 2017*, 2017.
- [Winata *et al.*, 2019] Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Jamin Shin, Yan Xu, Peng Xu, and Pascale Fung. CAiRE.HKUST at SemEval-2019 task 3: Hierarchical attention for dialogue emotion classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 142–147, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

Leveraging BERT to Improve the FEARS Index for Stock Forecasting

Linyi Yang¹, Yang Xu², Tin Lok James Ng³, Ruihai Dong¹

¹Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland

²Beijing Dublin International College, Beijing University of Technology, Beijing, China

³School of Mathematics and Applied Statistics, University of Wollongong, Australia

{linyi.yang, ruihai.dong}@insight-centre.org; yangxu@bjut.edu.cn; jamesng@uow.edu.au

Abstract

Financial and Economic Attitudes Revealed by Search (FEARS) index reflects the attention and sentiment of public investors and is an important factor for predicting stock price return. In this paper, we take into account the semantics of the FEARS search terms by leveraging the Bidirectional Encoder Representations from Transformers (BERT), and further apply a self-attention deep learning model to our refined FEARS seamlessly for stock return prediction. We demonstrate the practical benefits of our approach by comparing to baseline works.

1 Introduction

Efficient Market Hypothesis proposed by Fama [1965] states that stock market prices are driven by all observable information. In reality, it has been shown that investor sentiment can affect the asset prices due to the well-known psychological fact that investors with positive (negative) sentiment tend to make overly optimistic(pessimistic) judgments and decisions [Keynes, 1937]. These two classic theories open the gate of financial forecasting.

More recently, numerous empirical studies also provide consistent evidence to support the theory that investor sentiment has a significant impact on asset prices [Barber and Odean, 2007; Yuan, 2015; Da et al., 2011, 2014]. For instance, Peng and Xiong [2006] show that limited investor attention leads to category-learning behavior, i.e., investors tend to process more market-wide information than firm-specific information. Baker and Wurgler [2006] present evidence that the investor sentiment has effects on stock price movements across different stocks. They construct a novel investor sentiment index (BW index hereafter), and find that high investor sentiment predicts strongly low returns in the stock market. Vozlyublennai [2014] investigates a link between the performances of several security indices in broad investment categories with the exception of exchange rates. He finds a significant short-term change in index returns following an increase in attention. By analyzing the effect of attention on stock market, Yuan [2015] demonstrates that investor atten-

tion is one of the factors that inherently causes individual investors in aggregate to alter their stock positions dramatically, and he also suggests that the findings have implications for other research in finance.

In an influenced study, Da et al., [2014] build a list of search terms based on Google search volume in the U.S. market for various keywords that reveal sentiment toward economic conditions. By constructing a Financial and Economic Attitudes Revealed by Search (FEARS) index as a measure of investor sentiment, they find that "FEARS" is able to predict both short-term return and temporary increases in volatility. Tetlock [2007] shows that negative terms in English language are more useful for identifying sentiment compared to positive words. For this reason, the list consists of thirty negative search terms derived from words of economic sentiment in the Harvard and Lasswell dictionary [Tetlock, 2007] which have had the largest negative correlation with the market. The constructed list includes terms such as "gold prices" and "recession", which historically have had the largest daily correlation with the stock market. Finally, the FEARS index is defined by simply aggregating the change of each term's search volume, which implies that each term contributes equally to the FEARS index.

However, it may not be appropriate to assume that each of the search terms has the same level of contribution to stock market forecasting. Since previous works have not taken into account the semantics of the search terms in modelling their effects on the price movements. Moreover, the fluctuation of the volume of a search term may have a different effect on stock price movements on different days due to the complex dynamics of financial markets. Therefore, we argue that the current method of calculating the index is far from optimal. In this paper, instead of calculating index by simply aggregating the change of the thirty terms, FEARS index is refined by allocating different weights to different terms while the contribution is dynamic with the change of market.

In a nutshell, investor attention has been corroborated to be statistically and economically significant in security markets, while little research has been undertaken in the influence of the semantic information. To under the meaning of the search terms, Natural Language Processing (NLP) is leveraged. The

first key component in neural language understanding models is to find an approach to mathematically model words. A traditional method for representing words is the one-hot representation, where each word is represented as a binary vector with all but one entries of the vector are zero. Each integer value is represented as a binary vector that is all zero values except the index of the target word. However, there are two main shortcomings associated with such representations. First, the dimension of a vector increases accordingly when the number of words. Second, any two words represented by one-hot representation are isolated and cannot capture the information between words at the semantic level. In comparison, the use of a pre-trained word embedding allows clustering of similar words in a latent space, where semantically similar words are closer in the latent space. In recent years, language model pre-training has shown to be beneficial for improving downstream tasks of NLP [Peters *et al.*, 2017, 2018; Radford *et al.*, 2018; Howard and Ruder, 2018].

Various extensions to word embedding have been proposed. For example, ELMo [Peters *et al.*, 2018] which is short for Embeddings from Language Models representation differ from traditional word embedding in that each token is assigned a representation, task-specific models are used to include the pre-trained representations as additional features. Besides, the Generative Pre-Trained Transformer (OpenAI GPT) [Radford *et al.*, 2018] introduces a novel idea which involves fine-tuning the pre-trained parameters by jointly estimating task-specific parameters for the downstream tasks.

However, [Devlin, *et al.*, 2018] argue that the current techniques severely restrict the power of the pre-trained word representations. To address the limitation that the standard language models are unidirectional, Google improves the previous models of pre-training by proposing BERT: Bidirectional Encoder Representations from Transformers [Devlin, *et al.*, 2018]. They address the unidirectional constraints by proposing a new pre-training objective: the “masked language model” (MLM). Experimental results show that pre-trained representations eliminate the needs of many traditional heavily engineered task-specific models. It is one of the most representative works recently which can be seen as a milestone in the field of pre-training for language understanding.

By leveraging the pre-trained word embedding, many recent works have applied NLP techniques with multiple textual data sources to predict stock price movement [Si *et al.*, 2013; Ding *et al.*, 2014, 2015; Xu and Cohen, 2018]. Existing deep neural network approaches for stock price prediction have two main shortcomings. First, most of the proposed methods have focused on binary classifications of stock price movement (up or down). However, binary classification is less useful in the context of investment and financial risk management. To address this shortcoming, our developed methodology allows prediction of the return of a stock. Second, existing methods [Ding *et al.*, 2015, Xu and Cohen, 2018, and Feng *et al.*, 2018] typically employ the traditional approach in splitting the dataset into training and test sets in machine

learning, whereby the first k% of the data is allocated to the training set, and the remaining to the test set. However, such approach is not suitable for predicting stock market return, as the financial market may encounter structure changes. Hence, we adopt the recursive forecast, which is a common method in finance [Han *et al.*, 2018; Huang *et al.*, 2017; Rapach *et al.*, 2013].

In this paper, we first improve the construction of the FEARS index which represents the investor sentiment in order to get different input representations of search terms that integrating the semantic information. Then, we propose a self-attention neural network to predict the stock return using recursive training method.

The contributions of our papers are as follows:

- We propose a self-attention neural network with semantic information to predict the next short-term stock return and outperform the baseline works that only use financial index. To the best of our knowledge, semantic fears index is the first attempt to integrate semantic information with FEARS.
- We illustrate the importance of using semantic information for FEARS index to allocate different weights to different search terms.
- Unlike Si *et al.* [2013], Ding *et al.* [2014, 2015], and Xu and Cohen, [2018], we use recursive training for model estimation and prediction rather than the traditional way of splitting data into train and test sets.

2 Methodology

We introduce our model and its detailed implementation in this section. First, we provide an overview of the model architecture and the input representations. Then, we introduce our prediction model and the core innovation in this paper. Finally, the differences between our model and the classical model [Da *et al.*, 2014] are discussed in section.

2.1 Overview

The goal of our work is to leverage semantic information to improve FEARS for stock forecasting. To verify the performance of our refined FEARS index, a stock return predictive model is built in this paper. The previous state-of-the-art methods in text-based stock prediction connect the encoder and decoder through attention mechanisms [Si *et al.*, 2013; Ding *et al.*, 2014, 2015]. Hence, the Transformer network architecture [Vaswani *et al.*, 2017] proposed by Google, based solely on attention mechanisms is adopted in this paper for predicting the stock return. The Transformer is also known as self-attention mechanism.

Inspired by Vaswani *et al.*, 2017], we refine the FEARS index proposed by and [Da *et al.*, 2014] and test its efficiency in the task of stock return prediction. The overview of our model is shown in Figure 1. In general, our model contains four components:

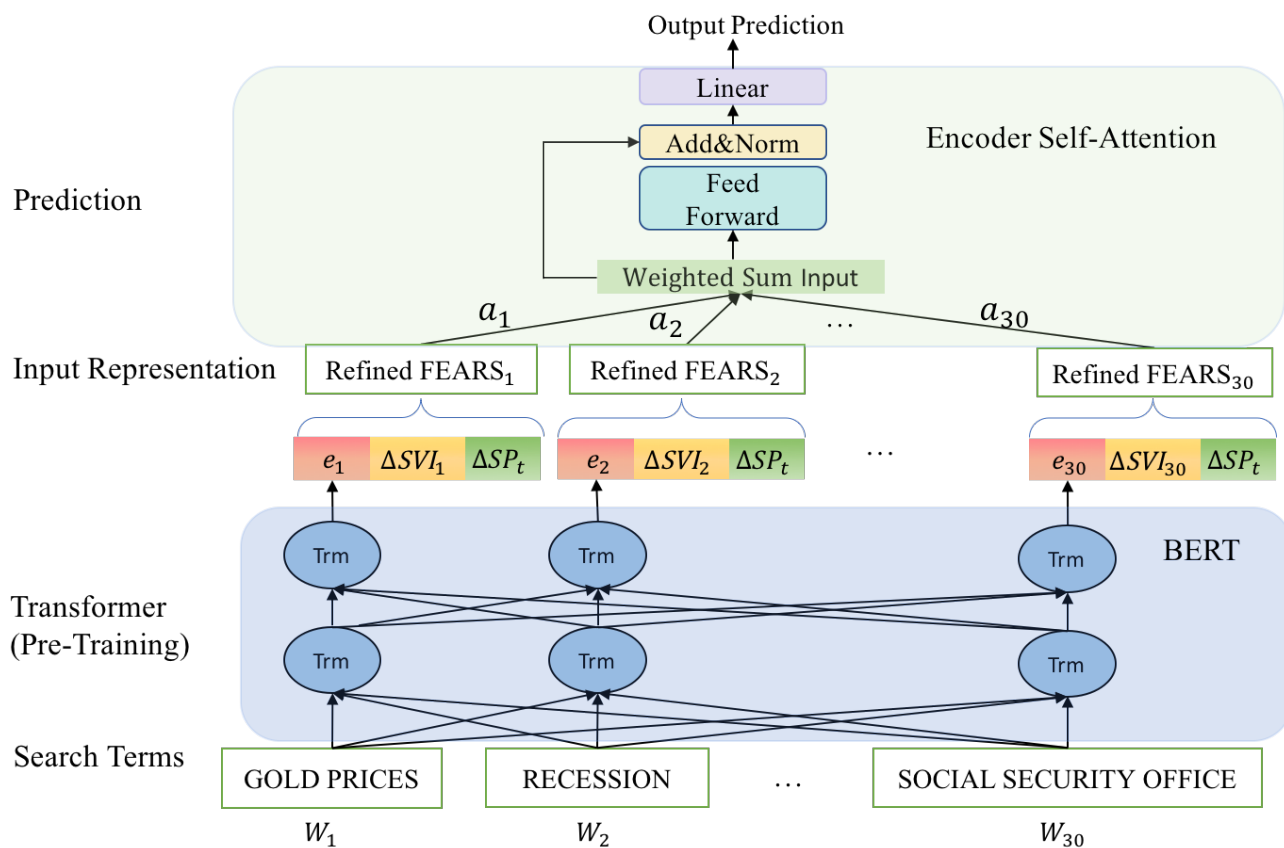


Figure 1: The architecture of our self-attention model for stock return prediction. The Pre-training model and input representation will be detailed introduced later in Section 2.2 and Section 2.3 respectively.

- 1) Data Source: Previously, Da [et al., 2014] observe the stock return has the strong negative effect associated with investor sentiment that can be presented by FEARS and last week’s stock return because of the return reversals in a short-term. Hence, our input contains both the FEARS and last week’s stock return.
- 2) Embedding layer: First, the thirty-selected search terms [Da et al., 2014] that have the largest negative influence for the market are pre-trained into embeddings based on the architecture of BERT. As a result, semantically similar search terms are mapped into similar locations in a latent space.
- 3) Input Representation: We obtain our input representation by combining the embeddings of the search terms with stock return and change in search volume.
- 4) Prediction Layer: The self-attention mechanism is used to allocate different weights to different search terms for stock return prediction. We then output the prediction of the next week’s stock return using a dual-layer feed forward neural network and the mean squared error is used as loss function to train the entire model.

Our model contains a bidirectional Transformer encoder-based on the original implementation described in [Vaswani et al., 2017] and a self-attention prediction model. We will first introduce the transformer model for word embedding, then cover the model for prediction using self-attention mechanism.

2.2 Transformer for search terms embedding

Firstly, BERT is used as a term encoding service to map our variable-length selected search terms to a fixed-dimension vector. BERT is a language representation model developed by Google. It leverages an enormous amount of public textual data on the web and is trained in an unsupervised fashion. Pre-training a BERT model is fairly expensive and time-consuming process. Hence, in this paper, a pre-trained model that contains 110M parameters is used to obtain representations of our search terms. The pre-trained model can be downloaded from Google¹.

We use BERT as a terms encoder and hosts it as a service via ZeroMQ [Hintjens, 2013] to map our search terms into fixed-dimension vectors $E = \{e_1, e_2, \dots, e_{30}\} \in \mathbb{R}^{30 \times D}$, where D is the dimension of the phrases embedding of the search terms

¹ <https://github.com/google-research/bert#pre-trained-models>

in the current timestamp, and the length of search terms is 30. We apply PCA [Jolliffe, 2011] to the embedded vectors in order for visualization, and the results are shown in Figure 2.

As a preliminary step, we examine if the word embeddings obtained from the pre-trained model can reasonably represent the semantic relatedness of the words. Following [Da et al., 2014], we just select the most influenced 30 negative search terms without any positive terms. Since in Tetlock (2007) it appears that negative terms in English language are most useful for identifying sentiment. As illustrated in Figure 2, terms with similar economic interpretations are closer in the projected two-dimensional space, and vice versa. We observe the clustering of the search terms “gold”, “gold prices” and “price of gold” that are all related to the precious metal gold, which is normally perceived as “safe heaven” of the capital market. Intuitively, capital inflows to gold market dramatically increase when equity markets experience bearish condition.

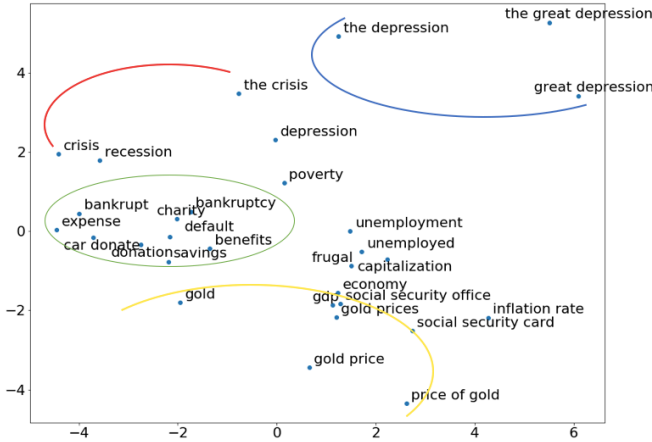


Figure 2: The visualization of our thirty selected search terms after embedding.

2.3 Input Representation

Da et al. [2014] found strong negative association between FEARS index and S&P 500 index daily return and claim that FEARS can be used as a proxy for investor sentiment. The calculation of FEARS involves only averaging the change in the search volume of the thirty selected terms. However, as explained in the previous section, such approach ignores the semantics of the search terms which may result in inferior predictions.

To address this issue, we propose a novel method to refine FEARS index by integrating the semantic information with the original calculation. Consequently, different weights will be assigned to each of the search term separately using a self-attention mechanism. In addition, the weights will be dynamically adjusted as the financial market evolves over time. We define the FEARS index corresponds to search term i term on day t as:

$$FEARS_{i,t} = e_i \times (\alpha \Delta SVI_{i,t} + (1 - \alpha) \Delta SP_t) \quad (1)$$

where e_i is the term embedding trained by BERT $e_i \in \mathbb{R}^{768}$, $\Delta SVI_{i,t}$ represents the weekly change in search volume for the search term i :

$$\Delta SVI_{i,t} = (SVI_{i,t} - SVI_{i,t-1}) / SVI_{i,t-1} \quad (2)$$

similarly, the ΔSP_t is the S&P500 index return in trading day t , it can be calculated as:

$$\Delta SP_t = (SP_t - SP_{t-1}) / SP_{t-1} \quad (3)$$

Finally, we generate our input representations in every timestamp to predict the next timestamp’s S&P500 return later:

$$FEARS = \{FEARS_1, FEARS_2, \dots, FEARS_l\} \quad (4)$$

where $FEARS \in \mathbb{R}^{l \times 30 \times 768}$, and l , 30, and 768 represent the length of the dataset, the thirty selected terms and the dimension of the latent space, respectively.

We adjust the fine-tuning procedure of the original BERT model [Devlin, et al., 2018] for our prediction task. Different from the original model where all parameters of BERT and the additional layer are fine-tuned jointly to minimize the loss, we fix the pre-trained word embeddings where are used to form our input representation. Keeping the embeddings fixed allows us to speed up our model training. Besides, we deprecate the position embedding since our input is not a sentence, in contrast to classical NLP tasks.

2.4 Predictive Model

The calculation of the prediction stock return r is shown below:

$$u_{t,i} = \text{ReLU}(W \cdot FEARS_{t,i} + b) \quad (5)$$

$$a_{t,i} = \exp(u_{t,i} u^T) / \sum_{i'=1}^{30} \exp(u_{t,i'} u^T) \quad (6)$$

$$v_t = \sum_{i=1}^{30} (a_{t,i} FEARS_{t,i}) \quad (7)$$

$$r_t = \text{FFN}(v_t) \quad (8)$$

where $u \in \mathbb{R}^{30 \times h}$ is the query vector, and equals to key and value vector in the self-attention mechanism; h denotes the number of hidden units; the weight matrix $W \in \mathbb{R}^{D \times h}$, D is the dimension of input; $v_t \in \mathbb{R}^D$ is the weighted sum of inputs. FFN is the short for feed forward neural network.

The use of a self-attention mechanism allows allocating different weights to the words when making out of sample predictions, it decides which word should be paid more attention by calculating the similarity between the query vector and the key vector. Then multiply the value vector with the score of the similarity after softmax. In the encoder self-attention mechanism, the query vector, key vector and value vector are

all itself. We adopt the dropout strategy for training our model. Finally, we get our prediction for the return of S&P500 index with a dual-level FFN in this paper.

2.5 Recursive Training

The standard approach [Ding et al. 2015; Xu and Cohen, 2018; and Yang et al. 2019] in splitting the data into training and test sets will generally not work well in financial applications, due to the difficulty in predicting stock returns using data from years ago. On the other hand, updating trading strategy periodically is a common approach in quantitative finance [Han et al., 2018; Huang et al., 2017; Rapach et al., 2013].

Hence, we apply the online-learning (recursive training) method to build our model. The parameters will be updated with the loss in last training period. In this paper, we set the training length as 8 weeks while the testing length is 4 weeks. It means that we make use of the last 8 weeks’ data to train our model while the next 4 weeks’ data for testing. The stock return of S&P500 will be recursively predicted by repeating this process.

3 Experiments

Our experiments aim to demonstrate that the semantic information integrating with search volume is beneficial to predict the stock return. In this section, we first introduce the procedure of the collecting the weekly search index and S&P500 return. Secondly, we discuss the loss function used in this paper. Next, we will specify the hyper-parameters in Section 3.3. Finally, we compare the performance of our method on S&P 500 index prediction to demonstrate the effect of self-attention mechanism with semantic information.

3.1 Data Collection

We use S&P 500 market index as the proxy for the US equity market and the historical prices are obtained from Quandl². Stock returns are computed as the change in end-of-week settlement prices.

To construct FEAR index for the US stock market, we use the public Search Volume Index (SVI) from Google Trends³ as attention proxies, following Da et al. (2014) and Han et al. (2018). The numbers present search probabilities of a given keyword at a given time. We consider the 30 terms that have been proven to be effectively associated with security prices from Da et al. (2014). These terms are suggested to contain information on financial markets and useful to predict future stock prices. All attention data cover a weekly period of 2004:01-2015:12. We work in logarithms of search terms probabilities for ease of exposition and notation.

3.2 Evaluation Metric

We use the Mean Square Error (MSE) to evaluate our model in stock return prediction. MSE is calculated as:

$$\text{MSE} = \frac{1}{n} \times \sum_{t=1}^n (\mathbf{p}_t - \widehat{\mathbf{p}}_t) \quad (9)$$

Where n denotes the length of total test sets, \mathbf{p}_t is the true value of the S&P500 index while $\widehat{\mathbf{p}}_t$ represents the output of our model at timestamp t .

3.3 Experiment Setup

Hyper-parameters for BERT. The hyper-parameters are shown in Table 1 and are the same as in the model BERT-BASE.

Settings	
Embedding Dimension	768
Number of Layers	12
Hidden Size	768
Attention Heads	12

Table 1: BERT-Base Hyper-parameters

Hyper-parameters for Prediction. Since the BERT-Base model we applied has 110M parameters. Hence, we change the terms embedding to non-trainable variables in our model. That is, we train our two models separately in order to speed up the training process. Experimental hyper-parameters of the prediction model are shown in Table 2.

Settings	
Input Size	768
Number of Layers	2
Hidden Units	256
Epochs	6
Batch Size	2
Optimizer	Adam
Dropout Probability	0.6

Table 2: Hyper-parameters of Prediction Model

3.4 Experimental Results

In this section, we demonstrate the efficiency of our proposed model based on our experimental results. We first reproduce the baseline work of [Da et al., 2014], then we compare different ways of integrating the semantic information with the baseline work in terms of their performance on the weekly dataset we collected. We evaluate our model using the online-training strategy. Since there are no previous attempts on adopting non-linear method based on the FEAR index, we just compare our method with the original strategy proposed by [Da et al., 2014] in experiments.

Baseline:

- FEARS and Asset prices [Da et al., 2014]: They use daily Internet search volume from millions of households to reveal market-level sentiment. Then

² <https://www.quandl.com>

³ <http://www.google.com/trend>

the volume of queries in U.S. are simply aggregated to construct FEARS. They finally use FEARS to predict short-term stock return by linear regression.

Our Method:

- We propose a novel model that integrates semantic information with the traditional financial index to predict the return of S&P500 index.

We test our methods and the baseline model using recursive forecast. The experimental results are shown in Table 3.

Method	MSE
Linear Regression (FEARS) [Da et al. 2014]	0.001094
Linear Regression (Optimal $\alpha = 0.6$)	0.000809
Transformer (FEARS)	0.001034
Transformer (Embedding $\times \Delta SP$; $\alpha = 0$)	0.000941
Transformer (Embedding $\times \Delta SVI$; $\alpha = 1$)	0.000678
Transformer (Optimal $\alpha = 0.6$)	0.000585

Table 3: Prediction Model

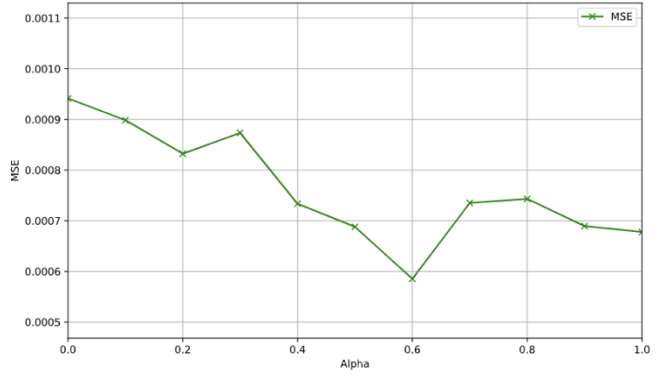
Since stock return prediction is a challenging task and a minor improvement usually leads to large potential profits. From Table 3, we can find that our model outperforms the baseline work in terms of the MSE loss.

Effects of Terms Embedding. First, in comparison with the performance of two works using linear regression model, we find that the MSE decreases if we construct the input representation with embedding. In addition, best predictive performance cannot be attained if only weekly aggregated search frequency (SVI) is used. These demonstrate the benefits of including the semantic information in the model, especially, the embedding of the search terms.

Effects of Self-Attention Mechanism. Second, the first two baseline methods simply take the sum of first search terms’ values as input which are unable to capture the fact that they have different contributions on different days. The performance of the proposed model with self-attention mechanism show that transformer model architecture is useful to predict stock return. It is mainly because our proposed model can allocate different weights to different search terms in terms of their importance for prediction on different trading days during the online-training and testing.

Finally, inspired by two main conclusions in [Da et al. 2014], namely, 1) FEARS has negative effect on the market, 2) short-term stock return predictability is reflected in the contrarian effect, we investigate the relative importance of the change in FEARS and last week’s S&P500 return on the performance of prediction.

The parameter α defined in Eq. (1) represents the tradeoff between the FEARS index and last week’s stock return. The



performance of the stock return prediction with a range of values for α is shown in Figure 3.

Figure 3: Performance of S&P500 weekly return prediction with varied α , see Eq. (1).

As shown in Figure 3, the best performance is achieved at $\alpha = 0.6$. The MSE loss curve gradually decreases as α increases, before reaching its minimum at $\alpha = 0.6$. It then ascends abruptly. The MSE loss curve finally remained relatively stable towards $\alpha = 1$.

4 Conclusion

This paper proposes a novel method for refining the FEARS, which can leverage the embedding of search terms to better represent the investor sentiment. Also, a prediction model based on self-attention mechanism is introduced for stock return prediction. It aims to automatically allocate different weights to different search terms considering their contribution to the target trading day. The experimental results on our weekly dataset illustrate that the semantic information benefits the task of stock return prediction, while a trade-off between the price data and search volume data is useful to improve the performance.

In the future, there are two potential extensions of this work: 1) The dictionary of top 30 search terms is fixed in this work. It might be beneficial to dynamically update the search terms used for prediction of stock return for capturing some fresh significant keywords. 2) The trade-off parameter α now is fixed at 0.6 in this work, by allowing α to vary across time, we may achieve better performance at stock return prediction.

Acknowledgments

This research was supported by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

- [Keynes, 1937] The general theory of employment. *The quarterly journal of economics*, 51(2): 209-223, 1937.
- [Fama, 1965] The behavior of stock-market prices. *The Journal of Business*, 38(1): 34-105, 1965.
- [Barber and Terrance, 2007] All that glitters: The effect of attention and news on the buying behavior of individual and institutional investors. *The review of financial studies* 21, no. 2: 785-818, 2007.
- [Yuan, 2015] Market-wide attention, trading, and stock returns. *Journal of Financial Economics*, 116(3): 548-564, 2015.
- [Da et al., 2011] In search of attention. *The Journal of Finance*, 66(5): 1461-1499, 2011.
- [Vozlyublennaiia, 2014] Investor attention, index performance, and return predictability. *Journal of Banking & Finance* 41: 17-35, 2014.
- [Baker and Wurgler, 2006] Investor sentiment and the cross-section of stock returns. *The journal of finance*. 61(4): 1645-1680, 2006.
- [Da et al., 2014] The sum of all FEARS investor sentiment and asset prices. *The Review of Financial Studies*, 28(1): 1-32, 2014.
- [Tetlock, 2007] Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, 62(3): 1139-1168, 2007.
- [Peters et al., 2017] Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the 2017 Association for Computational Linguistics (Volume 1: Long Papers)*, pages: 1756-1765, 2017.
- [Peters et al., 2018] Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages: 2227-2237, 2018.
- [Radford et al., 2018] Improving language understanding with unsupervised learning. Technical report, OpenAI, 2018.
- [Howard and Ruder, 2018] Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages: 328-339, 2018.
- [Devlin et al., 2018] Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [Si et al., 2013] Exploiting topic based twitter sentiment for stock prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages: 24-29, 2013.
- [Ding et al., 2014] Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages: 1415-1425, 2014.
- [Ding et al., 2015] Deep learning for event-driven stock prediction. In *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2015, pages: 2327-2333, 2015.
- [Xu and Cohen, 2018] Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages: 1970-1979, 2018.
- [Vaswani et al., 2017] Attention is all you need. In *Advances in neural information processing systems 30 (NIPS 2017)*, pages: 5998-6008, 2017.
- [Hintjens, 2013] ZeroMQ: messaging for many applications. *O'Reilly Media, Inc.*, 2013.
- [Jolliffe, 2011] Principal component analysis. *Springer Berlin Heidelberg*, 2011.
- [Yang et al., 2019] Explainable Text-Driven Neural Network for Stock Prediction. *arXiv preprint arXiv:1902.04994*, 2019.
- [Han et al., 2018] Forecasting the CNY-CNH pricing differential: The role of investor attention. *Pacific-Basin Finance Journal* 49: 232-247, 2018.
- [Huang et al., 2016] Forecasting stock returns in good and bad times: The role of market states. In *27th Australasian Finance and Banking Conference*, 2016.
- [Rapach et al., 2013] International stock return predictability: what is the role of the United States? *Journal of Finance* 4:1633-1662, 2013.

Economic Causal-Chain Search using Text Mining Technology

Kiyoshi Izumi and Hiroki Sakaji

School of Engineering, the University of Tokyo, Japan
izumi@sys.t.u-tokyo.ac.jp and sakaji@sys.t.u-tokyo.ac.jp

Abstract

In this research, we extract causal information from textual data and construct a causality database in the economic field. Furthermore, we develop a method to produce causal chains starting from phrases representing specific events and offer possible ripple effects and factors of specific events or situations. Using our method to Japanese textual data, we have implemented a prototype system that can display causal chains for user-entered words. A user can interactively edit the causal chains by selecting appropriate causalities and deleting inappropriate causalities. In this project, we will apply our method to English textual data such as financial news articles and financial reports. The economic causal-chain search algorithm can be applying to various financial information services.

1 Introduction

Economic news articles and financial reports contain various descriptions of cause and effect between economic factors such as price movements, product sales, employment, and trades. For example, “Hospital operator reconsiders London IPO because of Brexit uncertainty” and “the higher prices are likely to take a toll on manufacturers as well as consumers because the economy has decelerated greatly this quarter” appeared in Bloomberg Market News on March 21 2019.

It is beneficial to construct a database of economic causality and analyze the relationship between causality for both financial professionals and non-specialists. Such technology can support professionals’ report writing and businesses. For non-specialists, the technology can help them understand the implicit information about causal relationship behind the specialized texts.

It is, however, difficult to analyze the causality between economic phenomena only by the statistical analysis of numerical data. That is because human activities produce a causal relationship between economic phenomena. Human activities are determined by mental processes such as cognition, thinking, and emotion. Thus, economic causality is influenced by social and cultural situations. It is almost impossible to extract objective and universal causality by sta-

tistical analysis of numerical data like natural scientific phenomena.

2 Technical ideas

In this program, we analyze economic text data that seems to contain causality recognized by humans and construct a database of causality related to the economic field. Furthermore, we develop a method to search for causal chains derived from phrases representing specific events. Using this method, we implement a system that can display causal chains for user’s input words and select appropriate sequences or delete inappropriate sequences. Our method consists of the following steps.

1. Step 1 extracts sentences that include cause-effect expressions (causal sentences) from Japanese financial statement summaries using a support vector machine.
2. Step 2 obtains cause-effect expressions from the extracted sentences using syntactic patterns.
3. Step 3 constructs economic causal chains by connecting each cause-effect expression.

Step 1 and Step 2 are applied a method of [Sakaji *et al.*, 2017].

2.1 Step 1: Extraction of Causal Sentences

We developed a method for extracting causal sentences from economic texts. Since this method uses a support vector machine (SVM) for extraction, we will now explain how to acquire features from financial statement summaries. To extract causal sentences, our method uses the features shown in Fig. 1. We employ both syntactic and semantic features.

Syntactic features <ul style="list-style-type: none">• Pairs of particles
Semantic features <ul style="list-style-type: none">• Extended language ontology
Other features <ul style="list-style-type: none">• Part of speech of morphemes just before clue expressions• Clue expressions• Morpheme unigrams• Morpheme bigrams

Figure 1: Table example

We aim to use expressions that are frequently used in cause and effect expressions in sentences as syntactic features

2.2 Step 2: Extracting Cause-Effect Expressions

We employ a method by [Sakaji *et al.*, 2008] to extract cause-effect expressions using four syntactic patterns. We analyzed sentence structures and used a pattern matching method with syntactic patterns is shown in Fig. 2. In Fig. 2, “Cause” indicates a cause expression, “Effect” indicates an expression of effect and “Clue” indicates a clue expression.

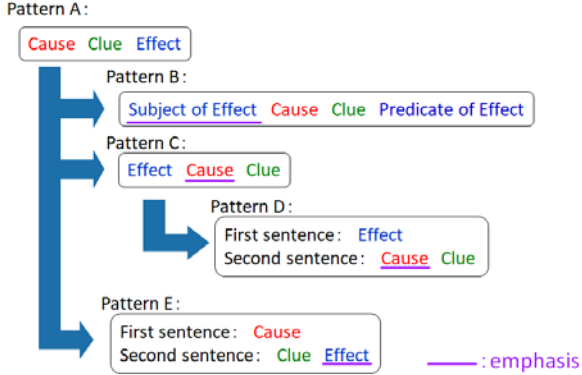


Figure 2: A syntactic patterns list

2.3 Step 3: Constructing Causal Chains

To construct causal chains, our method [Nishumura *et al.*, 2018] connects an effect expression of a causal expression and a cause expression of another causal expression. We show an algorithm of causal chains construction in Fig. 3.

In Fig. 3, “Company” indicates the company that issues the

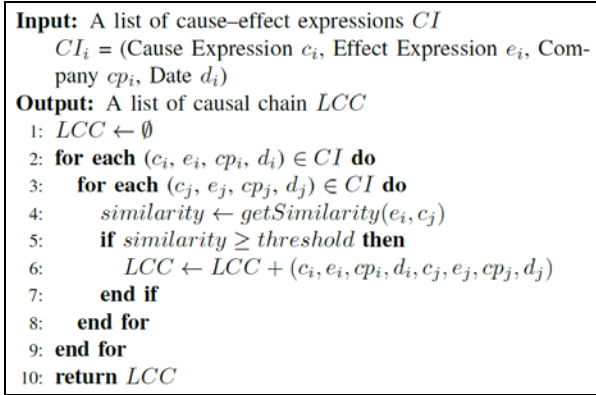


Figure 3: Construction of causal chains

financial statement summary from which the cause–effect expression has been extracted. Additionally, “Date” is the date the financial statement summary was issued. In Fig. 3, $getSimilarity(e_i; c_j)$ is a function that calculates similarity between the effect expression e_i and the cause expression c_j . Our method calculates the similarities based on vectors of word embedding. First, our method obtains word embedding

average of the words included in the expressions. Here, we define the average obtained from the effect expression e_i as \tilde{W}_{e_i} , and the average obtained from the cause expression c_j as \tilde{W}_{c_j} . Where $\tilde{W}_{e_i}, \tilde{W}_{c_j} \in R^m$ and m is the dimension size of word embedding. Then, our method calculates a cosine similarity between \tilde{W}_{e_i} and \tilde{W}_{c_j} , and employs the similarity as similarity between the effect expression e_i and the cause expression c_j . Finally, our method acquires pairs of cause-effect expressions as causal chain when the similarities are larger than a threshold.

3 Evaluation

In this section, we evaluate our method until Step 2. For evaluation, we use 30 pdf files of Japanese financial statement summaries and 30 documents of newspaper articles concerning business performance as test data. As results of human tagging, the 30 pdf files include 478 cause-effect expressions, the 30 documents include 51 cause-effect expressions. For classification of causal sentences, we use tagged 3,360 sentences that include 1,454 causal sentences. The tagger is an individual investor with 15 years of investment experience. We use MeCab (<http://taku910.github.io/mecab/>) for Japanese language morphological analyzer and CaboCha for Japanese dependency parser [Kudo *et al.*, 2002]. Moreover, we employ linear kernel as SVM kernel, and SVM^{Light} as SVM.

3.1 Evaluation Results

Table 1 shows experiment results. From Table 1, the method presents good performance for Japanese financial statement summaries and newspaper articles concerning business performance. Results of newspaper outperforms results of financial summaries. Because the method was developed for extracting cause-effect expressions from newspaper articles. However, results of financial summaries satisfy sufficient performance to construct causal chains. Therefore, we think that the method performance is enough to construct causal chains from financial texts.

Table 1: Evaluation results

	Precision	Recall	F1	Number of extracted expressions
Financial summaries	0.82	0.62	0.71	360
Newspaper	0.93	0.75	0.83	34

4 Prototype system

Based on the above-mentioned causal chain construction algorithm, the program of the basic framework of the economic causal chain search system for Japanese texts was implemented. You can try this system at <http://socsim.t.u-tokyo.ac.jp/ccs/>. Based on this system, we will develop English version of the sys-

tem. The behavior of this system is as follows. First, the user enters the start text (Fig. 4). The user can select the search direction, from cause to result or from result to cause. It is



Figure 4: Text box for entering start text

also possible to limit the search period of textual data. Click the search button to the right of the text box to display the causality chain from the input text (Fig. 5). By default, three causal relationships are displayed in descending order of similarities. If you want to see more causal relationships, you can click the "More" button to increase the display of causality nodes. If for each node of the causal relationship, the user determines that it is not appropriate, you can delete the node by pressing the delete button at the upper right of each node.

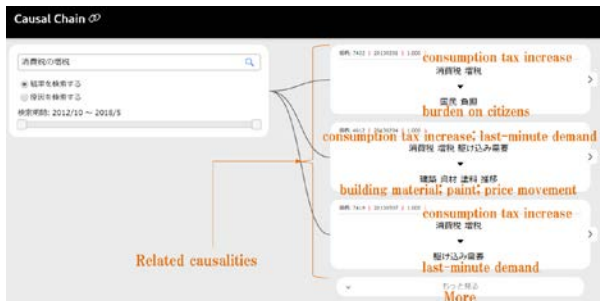


Figure 5: Display of causal relationships

If you want to further extend the causal chain from each node, click the ">" button on the right of each node, and the related causality is added with the clicked node as the terminal node (Fig. 6).

You can build the above-mentioned causal chain repeatedly and construct the causal chain required by the user, you can save the constructed causal chain in a file.

5 Application images

The current prototype system uses only small sized Japanese text, earnings summaries of Japanese firms. In order to improve the precision and recall of acquired causal chains, it is necessary to expand the text data. In this research program, we study the following two things to improve the causality database.

1. Expansion of a causal database using new text data such as news articles.



Figure 6: Extended causal chains

2. Extraction of causal information from English documents such as Form 10-k and press releases, and English database construction.

Our economic causal-chain search system and algorithm can be applied to various financial information services for both individual investors and financial professionals. Within the program period, we will implement the application service prototype program for any of the following application services. After the end of the program, we would like to launch some of the following services in collaboration with financial institutions or financial information vendors.

5.1 Services for Individual Investors

For non-specialists, the technology can help them understand the implicit information about causal relationship behind the specialized texts. One of the causes of this difficulty is the large gap between the knowledge of everyday life and that of finance. From everyday events to financial market trends, there is a causal-chain with financial specific knowledge. The proposed method can implement a service that provides knowledge to fill this gap.

(a) Presentation of background information in financial documents.

Using our algorithm, a user can search related stocks and possible factors derived from keywords and phrases in news articles and economic document-level (Fig. 7). By the influence search, a user can know which stocks' price may be affected by specific economic events and situations denoted in the documents. By the factor search, a user can know

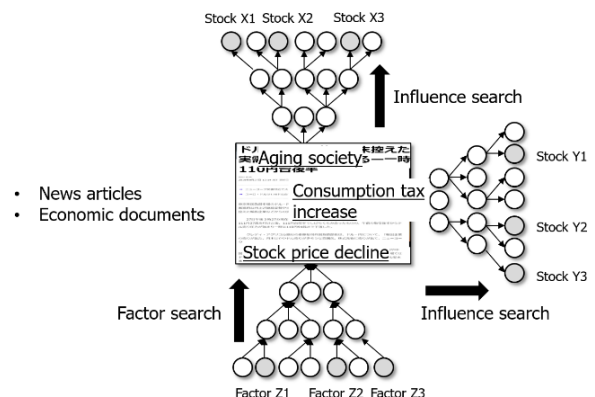


Figure 7: Influence and factor search from financial texts

possible causes of specific economic events and situations.

(b) Question Answering System.

Our algorithm can be applied to an interaction agent service provided by financial institutions for individual investors. Individual investors often want to ask basic questions to financial specialists and advisors. Because face-to-face advice from financial professionals is expensive, automated question answering leads to service penetration (Fig. 8).

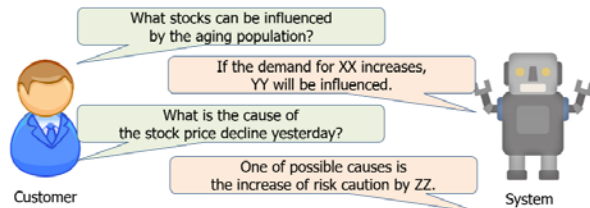


Figure 8: Question Answering System for Individual Investors

5.2 Supports for Financial Professionals Services

Our algorithm can be applied to a business support system for financial professionals in various departments of financial institutions such as market analysts and financial sales.

(a) Support for market report writing.

The proposed method can help market analysts decide the content when writing a report. For example, they search whether there is any influence from a certain event to the market to be explained, and decides whether this event should be written in the report. In addition, for a certain price movement, it is possible to search for potential factors and check whether there are any factors that should be written in the report.

(b) Sales support

Similar to the question answering system for individual investors mentioned above, when salespersons of a financial institution sell their financial products to a customer, they can search for stocks related to personal interests of the customer. If related stocks are searched in advance in relation to the interests of the customers, they can support sales activities. Also, for questions from customers, the above-mentioned question answering system can provide candidates for the contents to be answered.

6 Related work

Much work has been done on the extraction of causal information from texts. Inui et al. proposed a method for extracting causal relations (*cause*, *effect*, *precond* and *means*) from complex sentences containing the Japanese resultative connective “ため(*tame*: because)” [Inui et al., 2004], as this is a strong indicator of causal information. Khoo et al. proposed a method to extract cause-effect information from newspaper articles by applying manually created patterns [Khoo et al., 1998], as well as a method to extract causal knowledge from medical databases by applying their graphical patterns [Khoo et al., 2000]. Chang et al. proposed a

method to extract causal relationships between noun phrases using cue expressions and word pair probabilities [Chang et al., 2006], defined as the probability that the pair forms a causal noun phrase. Girju proposed a method for automatic detection and extraction of causal relations based on cue phrases [Girju, 2003] where causal relations are expressed by pairs of noun phrases. Girju used WordNet [Fellbaum, 1998] to create semantic constraints for selecting candidate pairs, so her method cannot extract unknown phrases that are not in WordNet. Bethard et al. proposed a method for classifying verb pairs that have causal relationships [Bethard et al., 2008] using an SVM for classification. Sadek et al. proposed a method for extracting Arabic causal relations using linguistic patterns [Sadek et al., 2016] represented using regular expressions. In contrast, our method not only extract cause-effect expressions but also construct causal chains.

Ishii et al. proposed a method for constructing causal chains using WordNet and SVO tuples [Ishii et al., 2012]. They employ method of [Sakaji et al., 2008] for extracting cause-effect expressions. Alashri et al. proposed a method to extract causal relations and construct causal chains from large text corpora related to climate change [Alashri et al., 2018]. However, their method can not construct causal chains when expressions consist of noun phrases. Because their method targets expressions that include Subjects, Verbs and Objects (SVO). On the other hand, our method is able to construct causal chains from expressions that consist noun phrases only.

7 Conclusions

We develop a method to produce causal chains starting from phrases representing specific events and offer possible ripple effects and factors of specific events or situations. Using our method to Japanese textual data, we have implemented a prototype system that can display causal chains for user-entered words. A user can interactively edit the causal chains by selecting appropriate causalities and deleting inappropriate causalities. In this project, we will apply our method to English textual data such as financial news articles and financial reports. The economic causal-chain search algorithm can be applying to various financial information services.

References

- [Alashri et al., 2018] Saud Alashri, Jiun-Yi Tsai, Anvesh Reddy Koppela, Hasan Davulcu: Snowball: Extracting Causal Chains from Climate Change Text Corpora, In the 2018 1st International Conference on Data Intelligence and Security (ICDIS), pages 234-241, 2018.
- [Bethard et al., 2008] Steven Bethard, James H.Martin: Learning Semantic Links from a Corpus of Parallel Temporal and Causal Relations, In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, pages 177-180, 2008.

- [Chang *et al.*, 2006] Du-Seong Chang, Key-Sun Choi: Incremental Cue Phrase Learning and Bootstrapping Method for Causality Extraction Using Cue Phrase and Word Pair Probabilities, *Information Processing and Management*, pages 662-678, 2006.
- [Fellbaum, 1998] Christiane Fellbaum: *WordNet: An Electronic Lexical Database*, The MIT Press, 1998.
- [Girju, 2003] Roxana Girju: Automatic detection of causal relations for Question Answering, In *ACL Workshop on Multilingual Summarization and Question Answering*, pages 76-83, 2003.
- [Inui *et al.*, 2004] Takashi Inui, Kentaro Inui, Yuji Matsumoto: Acquiring Causal Knowledge from Text Using the Connective Marker *tame*, *Journal of Information Processing Society of Japan*, pages 919-933, 2004.
- [Ishii *et al.*, 2012] Hiroshi Ishii, Qiang Ma, Masatoshi Yoshikawa, Incremental Construction of Causal Network from News Articles, *Journal of information processing*, pages 207-215, 2012.
- [Khoo *et al.*, 1998] Christopher S.G. Khoo, Jaklin Kornfilt, Robert N. Oddy, Sung Hyon Myaeng: Automatic Extraction of Cause-Effect Information from Newspaper Text Without Knowledge-based Inferencing, *Literary and Linguistic Computing*, pages 177-186, 1998.
- [Khoo *et al.*, 2000] Christopher S.G. Khoo, Syin Chan, Yun Niu: Extracting Causal Knowledge from a Medical Database Using Graphical Patterns, In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 336-343, 2000.
- [Kudo *et al.*, 2002] Taku Kudo, Yuji Matsumoto: Japanese Dependency Analysis using Cascaded Chunking, In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63-69, 2002.
- [Nishumura *et al.*, 2018] Kohei Nishumura, Hiroki Sakaji, Kiyoshi Izumi: Creation of Causal Relation Network using Semantic Similarity, the 32nd Annual Conference of the Japanese Society for Artificial Intelligence, p. 1P104, 2018.
- [Sadek *et al.*, 2016] Jawad Sadek, Farid Meziane: Extracting Arabic Causal Relations Using Linguistic Patterns, *ACM Transactions on Asian and Low-Resource Language Information Processing*, pages 1-20, 2016.
- [Sakaji *et al.*, 2008] Hiroki Sakaji, Satoshi Sekine, and Shigeru Masuyama: Extracting causal knowledge using clue phrases and syntactic patterns. In *7th International Conference on Practical Aspects of Knowledge Management (PAKM)*, pages 111–122, 2008.
- [Sakaji *et al.*, 2017] Hiroki Sakaji, Risa Murono, Hiroyuki Sakai, Jason Bennett, and Kiyoshi Izumi: Discovery of rare causal knowledge from financial statement summaries. In *The 2017 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFER)*, pages 602–608, 2017.

Transformer-Based Capsule Network For Stock Movements Prediction

Jintao Liu^{1*}, Xikai Liu^{1*}, Hongfei Lin^{1†}, Bo Xu^{1,2}, Yuqi Ren¹, Yufeng Diao^{1,3}, Liang Yang¹

¹Dalian University of Technology, Dalian, China

²State Key Laboratory of Cognitive Intelligence, iFLYTEK, P.R. China

³Inner Mongolia University for Nationalities, Tongliao, China

liujintao@mail.dlut.edu.cn, ws_lxk@mail.dlut.edu.cn, hflin@dlut.edu.cn, xubo@dlut.edu.cn
ryq13@mail.dlut.edu.cn, diaoyufeng@mail.dlut.edu.cn, liang@dlut.edu.cn

Abstract

Stock movements prediction is a highly challenging study for research and industry. Using social media for stock movements prediction is an effective but difficult task. However, the existing prediction methods which are based on social media usually do not consider the rich semantics and relation for a certain stock. It leads to difficulty in effective encoding. To solve this problem, we propose a CapTE (Capsule network based on Transformer Encoder) model which uses the Transformer Encoder to extract the deep semantic features of the social media and then captures the structural relationship of the texts through a capsule network. In this paper, we evaluate our method with different benchmarks, and the results demonstrate that our method improves the performance of stock movements prediction.

1 Introduction

According to the Efficient Market Hypothesis (EMH) [Fama *et al.*, 1969], stock price movements are thought to be related to the news. In natural language processing (NLP), public news and social media are two primary content resources for stock movements prediction. Moreover, social media such as Twitter is better in timeliness than news, so the condition that text from social media like Twitter is used to predict the stock movements draws numerous attention recently. On the other hand, tweets are able to reflect the investor’s mentality to some extent. It is useful for the prediction of stock movements.

Many studies focus on stock movements prediction based on social media. Xu and Cohen [2018] introduce recurrent and continuous latent variables for better treatment of stochasticity, use neural variational inference to address the intractable posterior inference, and also provide a hybrid objective with temporal auxiliary to flexibly capture predictive dependencies. Wu *et al.* [2018] propose a novel Cross-model attention based on Hybrid Recurrent Neural Network (CH-

RNN), which is inspired by the recent proposed DA-RNN model [Qin *et al.*, 2017].

In order to cross the chasm of prediction ability between machine and human, a deeper level of semantic information need to be explored in the text. Obviously, the above methods do not consider the rich semantic information and structural information of the social media, which lead to the model being unable to mine the deep semantics of text. For example, “Seems Travis is Still Running Uber! Surprised?”, this expression of the sentence is colloquial which is very difficult to get the real mentality of the author. Especially, for one stock, there are more than one tweets on the same day. Existing methods can not effectively extract semantic features from these complex texts which are vital for this task.

Therefore, insights on the solutions to stock movements prediction can be drawn from the novel structure for capturing a deeper level of semantic information. In this paper, we propose a CapTE (Capsule network based on Transformer Encoder) model which uses Transformer encoder to solve this problem due to its multi-head attention structure. The model is able to capture more important semantic information from different texts. Tang *et al.* [2018] prove that the Transformer encoder achieves better results in semantic feature extraction than other models based on CNN and RNN. From our results, we also prove that the Transformer encoder is better than other models in the task of stock movements prediction.

At the same time, there will be a lot of tweets for each stock on the same day. These tweets often contain different people’s views on the same stock, and the views are often different or even opposite. For example, “4 Major Stocks That Analysts Want You to Buy Now \$GE” and “\$GE technical alerts: Non-ADX 1,2,3,4 Bearish, MACD Bullish Signal Line Cross, 1,2,3 Retracement Be”. So how to capture valuable information from these different comments and ultimately get the right judgment is a very difficult problem. From the perspective of studying the relationship for different tweets contained on one day of one stock, we input the deep semantic information extracted by the Transformer encoder into a capsule network, which achieves the relationship between the semantic information for stock movements prediction. Ablation experiment proves that the capsule network effectively improves the accuracy of prediction. Finally, experimental results show that our integrated model is effective.

Our contributions are as follows:

*Both authors contributed equally to this paper

†Contact Author

- For the stock movements prediction, our model captures the deep effective semantic features of tweets more effectively through the Transformer encoder, compared with neural network models such as CNNs and RNNs.
- Up to date, no work introduces the Transformer to the task of stock movements prediction except us, and our model proves the Transformer improve the performance in the task of the stock movements prediction.
- The capsule network is also first introduced to solve the problem of stock movements prediction based on social media. The results show that the capsule network is effective for this task.

2 Related Work

Stock Market Prediction: There are a series of works predicting stock movements using text information [Lavrenko *et al.*, 2000; Schumaker and Chen, 2009; Xie *et al.*, 2013; Peng and Jiang, 2015; Li *et al.*, 2017]. Pioneering works extract different types of textual features from texts, such as bags-of-words, noun phrases, named entities, and structured events. Ding *et al.* [2014] showed structured events from open information extraction. Yates *et al.* and Fader *et al.* [2007; 2011] achieved better performance compared to conventional features, as they capture structured relations. However, one disadvantage of structured representations for events is that they lead to increased sparsity, which potentially limits the predictive power. Ding *et al.* [2015] proposed to address this issue by representing structured events and using event dense embeddings. Ding *et al.* [2016] leveraged ground truth from the knowledge graph to enhance event embeddings. Shah *et al.* [2018] retrieved, extracted, and analyzed the effects of news sentiments on the stock market. Liu *et al.* [2018] adopted a two-level attention mechanism to quantify the importance of the words and sentences in given news and designed a novel measurement for calculating the attention weights to avoid capturing redundant information in the news title and content. In this paper, we focus on capturing the deep semantic features of the social media appeared on the same day for prediction. The results show that our model is useful.

Transformer: Vaswani *et al.* [2017] presented the Transformer, the first sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention. Tang *et al.* [2018] evaluated RNNs, CNNs, and Transformer on two tasks: subject-verb agreement and word sense disambiguation. Their experimental results showed that: 1) Transformer and CNNs did not outperform RNNs in modeling subject-verb agreement over long distances; 2) Transformer performed distinctly better than RNNs and CNNs on word sense disambiguation. Radford *et al.* [Radford *et al.*, 2018] performed three different ablation studies and analyzed the effect of the Transformer by comparing it with a single layer 2048 unit LSTM using the same framework. They observed a 5.6 average score drop when using the LSTM instead of the Transformer. In our model, we obtain more semantic features through the Transformer. It is better than other baselines based on CNN and RNN.

Capsule Network: Hinton *et al.* [2011] firstly introduced the concept of “capsules” to address the representational limitations of CNNs and RNNs. Capsules with transformation matrices allow networks to automatically learn part-whole relationships. Consequently, Sabour *et al.* [2017] proposed capsule networks that replace the scalar-output feature detectors of CNNs with vector-output capsules and max-pooling with routing-by-agreement. The capsule network has shown its potential by achieving a state-of-the-art result on MNIST data. Xi *et al.* [2017] further tested out the application of capsule networks on CIFAR data with higher dimensionality. Hinton *et al.* [2018] proposed a new iterative routing procedure between capsule layers based on the EM algorithm, which achieved significantly better accuracy on the small NORB dataset. Zhang *et al.* [2018] generalized existing routing methods within the framework of weighted kernel density estimation. Zhao *et al.* [2018] investigated the performance of capsule networks in NLP tasks. In comparison, our work combines the transformer encoder with a capsule network, further improves the results of the task of stock movements prediction.

3 CapTE Model

We predict the movements of stocks on the trading day td . And we use price data crawled from Yahoo Finance to label the tweets that where 1 denotes rise and 0 denotes fall,

$$y = 1(p_{td}^c > p_{td-1}^c) \quad (1)$$

where p_{td}^c denotes the adjusted closing price which is adjusted for actions affecting stock movements, e.g. dividends and splits. Before our work, the adjusted closing price has been used for predicting stock price movements [Xie *et al.*, 2013] [Xu and Cohen, 2018].

Generally, tweets of the same stock often contain more than one item in the same trading day. For learning more valuable information from multiple tweets, we adopt the transformer to encode the texts and then get the encoded representation as the input of the capsule network. By the capsule network, we capture the relationship between different tweets appeared on the same trading day that belonging to one stock. Finally, we obtain the probability of each category as the prediction results. Figure 1 shows the architecture of the proposed model, namely Capsule network based on Transformer Encoder (CapTE).

We merge all the tweets appeared on one day of the same stock as one sentence. For each sentence s_i , we utilize the pretrained word embeddings (word2vec) to project each word token onto the d_{model} -dimensional space as the input of the Transformer encoder.

3.1 Transformer Encoder

In order to obtain deep semantic features from complex texts, we introduce the Transformer encoder. The encoder maps an input sequence of symbol representations $s_i = (x_1, \dots, x_n)$ to a sequence of continuous representations (z_1, \dots, z_n) . And as the paper [Vaswani *et al.*, 2017] design, the encoder contains a stack of $N = 6$ identical layers. Each layer has two sub-layers. A multi-head self-attention mechanism is the

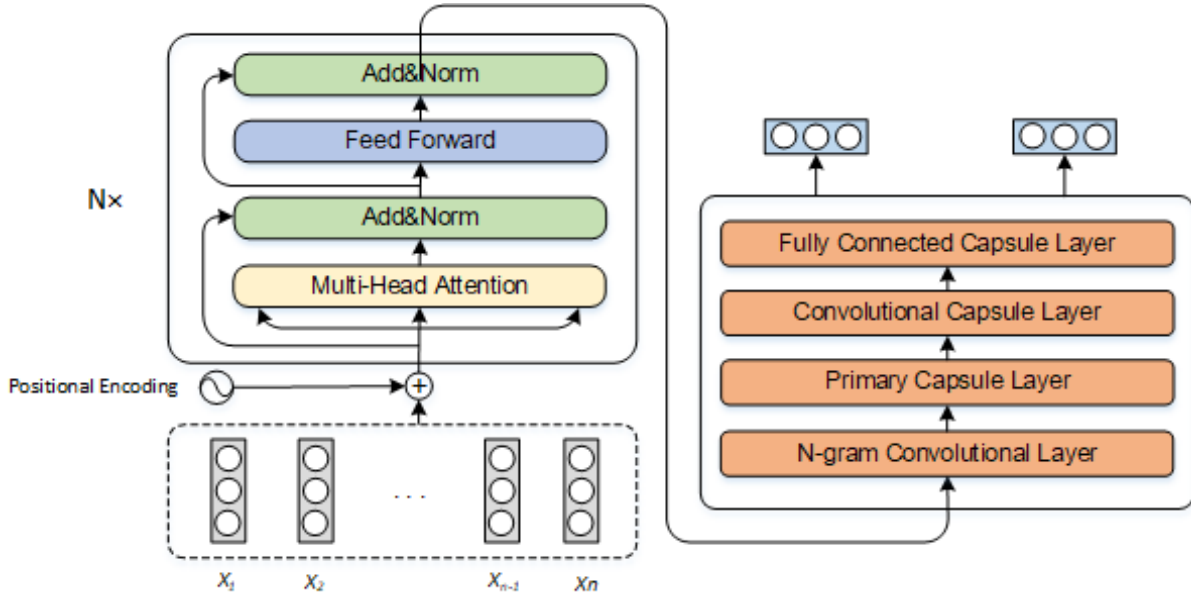


Figure 1: Transformer-Based Capsule Network

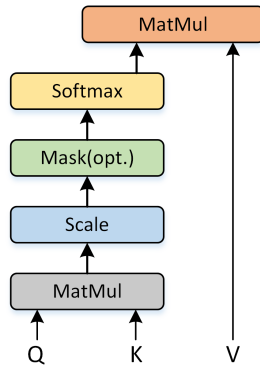


Figure 2: Scaled Dot-Product Attention.

first, while the second is a simple, position-wise fully connected feed-forward network. Around each of the two sub-layers exists a residual connection [He *et al.*, 2016], and a layer normalization follows after the two sub-layers. Hence, the output of each sub-layer is $\text{LayerNorm}(x + \text{Sublayer}(x))$. $\text{Sublayer}(x)$ is the function implemented by the sub-layer itself.

Positional Encoding

The order of the words in the tweets is significant for the prediction. A reversal of the order of the words in a sentence often changes the original meaning. For example, “breakout and buying” and “buying and breakout”. The former tweet means after breakout we can buy, but the later means we can buy immediately and wait for the raising of the stock price. So we adopt the “positional encodings” and add it to the input embeddings. In the end, we sum the two vectors as the final input at the bottom of the encoder. It is realized by the same dimension of the input embeddings and positional en-

codings. In our model, we employ sine and cosine functions of different frequencies as positional encodings:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (2)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (3)$$

where pos is the position and i is the dimension. Each dimension of the positional encoding corresponds to a sinusoid.

Scaled Dot-Product Attention

After obtaining the representation summed by the word embeddings and the positional encoding, we compute the matrixes as the input of Scaled Dot-Product Attention, which consists of queries and keys of dimension d_k , and values of dimension d_v . And then, we get the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and utilize a softmax function to achieve the weights on the values. The matrix Q can be packed together into a matrix after computing the attention function on a set of queries simultaneously. The matrices K and V also apply the same method, which denotes the keys and values respectively. And the whole process is depicted in Figure 2. The matrix of outputs are as follows:

$$\text{Attention}(Q; K; V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

Multi-Head Attention

For the complex tweets, a single attention function is difficult to achieve enough information for improving the result of the prediction. So we linearly project the queries, keys, and values h times with different, which learns linear projections to d_k , d_k and d_v dimensions, respectively. On each of these queries, keys, and values, we perform the attention function in parallel and yielding d_v -dimensional output values. They are concatenated and projected, resulting in the final values.

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. The function of Multi-Head Attention as follows:

$$MultiHead(Q; K; V) = Concat(head_1, \dots, head_h)W^O \quad (5)$$

where W^O is the weight matrix used to multiply with the concatenated result of all the heads to produce the final output of the encoder. In our work we adopt $h = 8$ parallel heads. For each of the heads, we use $d_k = d_v = d_{model}/h = 64$.

Position-wise Feed-Forward Networks

Each of the layers in the encoder contains a fully connected feed-forward network. It is applied to each position separately and identically and consists of two linear transformations with a ReLU activation in between.

$$FFN(x) = max(0; xW_1 + b_1)W_2 + b_2 \quad (6)$$

while the linear transformations are the same across different positions, they use different parameters from layer to layer, just as two convolutions with kernel size 1. The dimensionality of input and output is $d_{model} = 512$.

3.2 Capsule Network

Different from the method of CNNs and RNNs, the capsule network increases the weights of similar information through its dynamic routing. The dynamic routing is proposed by Sabour et al. [2017], which displaces the pooling operation used in conventional convolution neural network. It maintains the position information for features, which is beneficial to the text representation. More importantly, this routing-by-agreement method has the ability to cluster the features into each class. People often focus on the important event, hence, the comments based on a hot event look alike. Since the price of a stock is usually decided by the significant event, we choose the capsule network to handle the information obtained from the Transformer encoder. After getting the output from the transformer encoder, we input the new representation into a capsule network to get the probability of each category. The capsule network consists of four layers: n-gram convolutional layer, primary capsule layer, convolutional capsule layer, and fully connected capsule layer.

N-gram Convolutional Layer

This layer is a standard convolutional layer which extracts n-gram features at different positions of a sentence according to various convolutional filters. In this part, the sentence is the new representation from the Transformer encoder.

Suppose $Z \in R^{L \times d_{model}}$ denotes the input representation, where L is the length of the emerged tweets of a certain day. And $z_i \in R^{d_{model}}$ is the d_{model} -dimensional vector corresponding to the i th word in the new representation. $W^a \in R^{K_1 \times d_{model}}$ is the filter for the convolution operation, where K_1 is N-gram size. A filter W^a convolves with the word window $Z_{i:i+K_1-1}$ at each possible position (with stride of 1) to produce a column feature map $m^a \in R^{L-K_1+1}$, each element $m_i^a \in R$ of the feature map is produced by

$$m_i^a = f(Z_{i:i+K_1-1} \circ W^a + b_0) \quad (7)$$

where \circ is element-wise multiplication, b_0 is a bias term, and f is a nonlinear activate function. That is the process by which one feature is extracted from one filter. For $a = 1, \dots, B$, B filters with the same kernel size of the convolution operation. By assembling B feature maps together, we have a B -channel layer.

$$M = [m_1, m_2, \dots, m_B] \in R^{(L-K_1+1) \times B} \quad (8)$$

Primary Capsule Layer

The feature maps generated from the n-gram convolutional layer are fed into this layer, piecing the instantiated parts together via another convolution. This is the first capsule layer in which the capsules replace the scalar-output feature detectors of CNNs with vector-output capsules to preserve the instantiated parameters of each feature.

By sliding over the feature map M , each filter W^b output a series of capsules $p_i \in R^d$, where d is the dimension of the capsule. These capsules comprise a channel p_i of the primary capsule layer.

$$p_i = g(W^b M_i + b_1) \quad (9)$$

where g is nonlinear squash function through the entire vector, b_1 is the capsule bias term. For all C filters, the generated capsule feature maps can be rearranged as

$$P = [p_1, p_2, \dots, p_C] \in R^{(L-K_1+1) \times C \times d} \quad (10)$$

where totally $(L - K_1 + 1) \times C$ d -dimensional vectors are collected as capsules in P .

Child-Parent Relationships

Capsule network generates the capsules in the next layer using ‘‘routing-by-agreement’’. This process takes the place of pooling operation and usually discards the location information, which helps augment the robust of the network and cluster features for prediction. It allows the networks to automatically learn child-parent relationships. In the stock prediction task, different tweets with the same category are supposed to share a similar topic but with different viewpoints. For example, ‘‘4 Major Stocks That Analysts Want You to Buy Now \$GE’’ and ‘‘\$GE technical alerts: Non-ADX 1,2,3,4 Bearish, MACD Bullish Signal Line Cross, 1,2,3 Retracement Be’’. For the two comments, they talk about the same topic of technical analysis but get different views.

Between capsules i and j , a prediction vector $u_{j|i} \in R^d$ is first calculated from the child capsule i , by multiplying a weight matrix $W^{t_1} \in R^{N \times d \times d}$, where N is the number of parent capsules in the layer above. Each corresponding vote is computed by:

$$u_{j|i} = W_j^{t_1} u_i + b_{j|i} \in R^d \quad (11)$$

where u_i is a child-capsule in the layer below and $b_{j|i}$ is the capsule bias term.

The length of the capsule represents the probability that the input sample has the object capsule describes, and it is limited in range from 0 to 1 by a non-linear squashing function. The

function pushes the short vectors to shrink to zero length and the long ones to one.

$$v_j = \frac{\|g_j\|^2}{1 + \|g_j\|^2} \frac{g_j}{\|g_j\|^2} \quad (12)$$

A capsule g_j is generated by the linear combination of all the prediction vectors with weights.

Dynamic Routing

The basic idea of dynamic routing is to construct a non-linear map in an iterative manner ensuring. It shows that the output of each capsule gets sent to an appropriate parent in the subsequent layer [Sabour *et al.*, 2017]:

$$\{u_{j|i} \in R^d\}_{i=1, \dots, H, j=1, \dots, N} \mapsto \{v_j \in R^d\}_{j=1}^N \quad (13)$$

where v_j denotes each parent-capsule in the layer above. The parent capsules and their probabilities in the layer above are denoted as

$$v, a = \text{Routing}(\hat{u}) \quad (14)$$

where \hat{u} denotes all of the child capsules in the layer below, v denotes all of the parent-capsules and their probabilities $a = |v|$.

Convolutional Capsule Layer

In this layer, each capsule is connected with a local region $K_2 \times C$ spatially in the layer below. Those capsules in the region multiple transformation matrices are to learn child-parent relationships followed by routing by agreement to produce parent capsules in the layer above. $K_2 \times C$ is the number of child capsules in a local region in the layer above. When the transformation matrices are shared across the child capsules, we get each potential parent capsule. And then, we use routing-by-agreement to produce parent capsules feature maps totally $(n - K_1 - K_2 + 2) \times D$ d -dimensional capsules in this layer. D is the number of parent capsules which the child capsules are sent to.

Fully Connected Capsule Layer

The capsules in the layer below are flattened into a list of capsules and fed into fully connected capsule layer in which capsules are multiplied by transformation matrix $W^{d_1} \in R^{G \times d \times d}$ or $W^{d_2} \in R^{H \times G \times d \times d}$ followed by routing-by-agreement to produce final capsule v_j and its probability a_j for each category. And H is the number of child capsules in the layer below, $G = 3$ is the number of categories plus an extra orphan category in this task. The orphan category helps us collect the less contributive capsules that contain too much background information. This method reduces the interference for normal categories.

Inspired by Zhao *et al.* [2018], we attempt to use the probability of the existence of parent capsules to iteratively amend the connection strength. The length of the vector v_j represents the probability of each relation. To increase the difference between the lengths of categories, we adopt a separate margin loss $Loss_k$ for each relation capsule k :

$$Loss_k = Y_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - Y_k) \max(0, \|v_k\| - m^-)^2 \quad (15)$$

Data	Stocks	Days	Tweets	Words
Tweet	47	231	746,287	137,052

Table 1: Basic statistics of the dataset.

where v_k is the capsule for class k , m^+ and m^- is the top and bottom margins respectively. $Y_k = 1$ if the relation k is present. λ is the weight for the absent classes. In our model, $m^+ = 0.9$, $m^- = 0.1$ and $\lambda = 0.5$.

The Architectures of Capsule Network

The capsule network starts with a 1-gram ($K_1 = 1$) convolutional layer with 32 filters ($B = 32$) and a stride of 1 with ReLU non-linearity. All the other layers are capsule layers starting with a $B \times d$ primary capsule layer with 32 filters ($C = 32$), followed by a $1 \times C \times d \times d$ ($K_2 = 1$) convolutional capsule layer with 16 filters ($D = 16$) and a fully connected capsule layer in sequence.

Each capsule has 16-dimensional ($d = 16$) instantiated parameters and their length (norm) describe the probability of the existence of capsules. The capsule layers are connected by the transformation matrices, and each connection is also multiplied by a routing coefficient. It is dynamically computed by the routing of agreement mechanism. The final output with three classes ($G = 3$) in the fully connected capsule layer is obtained from the probability of each category. In this way, the capsule network learns more valuable information for the stock prediction.

4 Experiment

4.1 Datasets

We test our model on the open dataset¹. It ranges from January 2017 to November 2017 and contains 47 stocks which have sufficient tweets from the Standard Poor’s 500 list. The basic statistics of the dataset are shown in Table 1. The experimental dataset is still available until June 2019. Totally in our model and other baselines, we split the dataset with the ratio of approximately 5: 1: 1 in chronological order, which is the same as Wu *et al.* [2018].

4.2 Experimental Setups

In our experiment, the initial word embedding is obtained by word2vec. The dimension of word embedding is 512. We use the rise (1) and fall (0) of the stock price as the final output. The internal weights in our model are initialized by sampling from the uniform distribution and tuned in the training process. We adopt mini-batch in the training process, and the batch size is 128.

4.3 Evaluation Metrics

Following previous work for stock prediction [Xie *et al.*, 2013; Ding *et al.*, 2015; Xu and Cohen, 2018], we adopt the standard measure of accuracy and Matthews Correlation Coefficient (MCC) as evaluation metrics. With the confusion matrix which contains the number of samples classified as

¹<https://github.com/wuhuizhe/CHRRN>

Model	Acc.	MCC
TSLDA	53.92	0.0561
HAN	57.14	0.0723
HCAN	58.72	0.0876
CH-RNN	59.15	0.0945
CapTE-nT	59.64	0.1073
CapTE-nC	60.12	0.1258
CapTE	64.22	0.3481

Table 2: Performance of baselines and CapTE variations in accuracy and MCC.

true positive, false positive, true negative and false negative, MCC is calculated as follows:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \quad (16)$$

4.4 Comparison Methods

We conduct extensive experiments to compare our model with several baselines:

- TSLDA: A generative topic model jointly learning topics and sentiments [Nguyen and Shirai, 2015].
- HAN: A state-of-the-art discriminative deep neural network with hierarchical attention [Hu *et al.*, 2018]. In our experiment, we adopt their open source code² to get the results.
- HCAN: A novel deep generative model jointly exploiting text and price signals [Xu and Cohen, 2018].
- CH-RNN: A novel Cross-model attention based Hybrid Recurrent Neural Network (CH-RNN) [Wu *et al.*, 2018]. And the dataset is the same as our model.
- CapTE-nT: In order to verify the effectiveness of the Transformer, we conduct experiments by utilizing our model without the Transformer Encoder.
- CapTE-nC: In order to verify the effectiveness of the Capsule network, we conduct experiments by utilizing our model without the Capsule Network.

4.5 Experimental Analysis

We test our model from two aspects. One is to predict the rise and fall based on the dataset. The results are shown in table 2. The other is a simulation test based on the transaction. The results are shown in table 3. In the end, we analyze the reasons for error based on the different model.

Comparison Methods

From table 2, we can see that on the 47 stocks CH-RNN gets the highest score in the baselines. However, on the same dataset that using the same way to split, our model obtains higher accuracy than CH-RNN, more than 5%, that shows our model capture deep semantic features more effectively.

²<https://github.com/yumoxu/stocknet-code>

Stock	CH-RNN	CapTE
AAPL	884\$	901\$
BAC	872\$	996\$
DIS	659\$	869\$
IBM	1092\$	1768\$
PFE	1025\$	853\$
WMT	1127\$	1489\$

Table 3: Profit comparison between CH-RNN and CapTE.

The CapTE-nC model gets a higher score by using the only Transformer than CH-RNN in both accuracy and MCC. It further illustrates that the Transformer captures deep semantic features compared to RNNs. At the same time, the performance of CapTE-nT is higher than the scores of CH-RNN in both the accuracy and the MCC. It further demonstrates that the capsule network obtains valuable relationship information. On the other hand, the score of CapTE-nC is higher than CapTE-nT which indicates that the capture of deep semantic features is more important for complex data such as tweets.

Especially, the results of CapTE-nT and CapTE-nC with the only partial model are quite different from those of the complete model. We believe that it is because the transformer encoder and capsule network complement each other in the extraction of deep semantic features. They constitute a complete system. And the function of the system performs more effectively than a single model.

Stock Trading Simulation

We simulate real stock trading by following the strategy proposed by Lavrenko *et al.* [2000]. If the model predicts that a stock price will raise the next day, we spend \$10,000 to buy it at the opening price. And then, we hold the stock for one day. During this time, if the stock price increase 2% or more, we sell it immediately. If not, we sell the stock at the closing price. On the other hand, if the model predicts that a stock price will fall, when we can buy the stock at a price 1% lower than shorted, we buy the stock. Otherwise, we buy the stock at the closing price.

In table 3, we show the returns of six randomly selected stocks through CH-RNN and CapTE with \$10,000 in 20 trading day. And the maximum return of IBM is over 17%. The results demonstrate consistently better performance, which indicates the robustness of our model.

Error Analysis

2017/7/16	PFIZER...waiting for breakout and buying \$PFE
2017/11/10	RT @Focus_on_Dvds: Significant Insider Trades: Oct 30 - Nov 3

Figure 3: Examples for Error Analysis.

However, comparing with the PFE’s return between the

selected model, we find that CapTE is bad than CH-RNN. Hence, we compare the prediction results of CapTE with CH-RNN and analyze the cases which are wrongly predicted by CapTE but well predicted by CH-RNN.

Finally, we summarize two situations: a tweet is written to confuse the traders by the market makers. For example in Figure 3, “PFIZER...waiting for breakout and buying”, after this message, the stock price fell in the next few days. Instigating people to buy when they prepare to short and instigating people to sell when they want to bull is the main method to obtain profit by the market makers. Second, the event is just a fictive fact. For example, “Significant Insider Trades: Oct 30 - Nov 3”, and this message is apparently important to the stock trend. But it is well-known that the “Insider Trades” is unlikely to be made public. Most of the time, such news is just a rumor. For our model, it is hard to achieve the correct prediction without introducing the relevant knowledge in these conditions.

5 Conclusion

To capture the deep semantic information and structural relation for stock movements prediction task, we introduce the CapTE (Capsule network based on Transformer Encoder) model and demonstrate the reliability of our model. As shown in the results, we have no reason to doubt the importance of valuable information obtained through the Transformer. At the same time, with the aid of transformer encoder, the capsule network obtains the specific relationship between tweets that can improve the prediction accuracy of stock movements. Our model combines the advantages of the Transformer encoder and capsule network. In addition, because we introduce no financial data except texts in our model, our method has a generalization ability to the text classification tasks in the NLP field. However, our experimental dataset is only the day-level, the impact of tweets might be limited to the day when the event happens. Especially on the U.S. stock market, it allows people to trade many times on one trading day. For the task, this condition means the tweets have lost their impacts on the next day. Hence, how to predict the movements in a smaller period of time with information is the next topic we need to research.

Acknowledgments

This work is partially supported by a grant from the Natural Science Foundation of China (No.61632011, 61572102, 61702080, and 61602079), the Fundamental Research Funds for the Central Universities (No.DUT18ZD102, No.DUT19RC(4)016), the National Key Research Development Program of China (No.2018YFC0832101), the Post-doctoral Science Foundation of China (No.2018M631788, No.2018M641691), the Ministry of Education Humanities and Social Science Project (No.19YJCZH199), and the Foundation of State Key Laboratory of Cognitive Intelligence, iFLYTEK, P.R. China (COGOS-20190001, Intelligent Medical Question Answering based on User Profiling and Knowledge Graph).

References

- [Ding *et al.*, 2014] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1415–1425, 2014.
- [Ding *et al.*, 2015] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [Ding *et al.*, 2016] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Knowledge-driven event embedding for stock prediction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2133–2142, 2016.
- [Fader *et al.*, 2011] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1535–1545. Association for Computational Linguistics, 2011.
- [Fama *et al.*, 1969] Eugene F Fama, Lawrence Fisher, Michael C Jensen, and Richard Roll. The adjustment of stock prices to new information. *International economic review*, 10(1):1–21, 1969.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [Hinton *et al.*, 2011] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.
- [Hinton *et al.*, 2018] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018.
- [Hu *et al.*, 2018] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 261–269. ACM, 2018.
- [Lavrenko *et al.*, 2000] Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining*, volume 2000, pages 37–44, 2000.
- [Li *et al.*, 2017] Qing Li, Jun Wang, Feng Wang, Ping Li, Ling Liu, and Yuanzhu Chen. The role of social sentiment in stock markets: a view from joint effects of multiple information sources. *Multimedia Tools and Applications*, 76(10):12315–12345, 2017.
- [Liu *et al.*, 2018] Qikai Liu, Xiang Cheng, Sen Su, and Shuguang Zhu. Hierarchical complementary attention network for predicting stock price movements with news. In *Proceedings of the 27th ACM International Conference*

- on *Information and Knowledge Management*, pages 1603–1606. ACM, 2018.
- [Nguyen and Shirai, 2015] Thien Hai Nguyen and Kiyooki Shirai. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1354–1364, 2015.
- [Peng and Jiang, 2015] Yangtuo Peng and Hui Jiang. Leverage financial news to predict stock price movements using word embeddings and deep neural networks. *arXiv preprint arXiv:1506.07220*, 2015.
- [Qin *et al.*, 2017] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.
- [Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/language_understanding_paper.pdf, 2018.
- [Sabour *et al.*, 2017] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.
- [Schumaker and Chen, 2009] Robert P Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12, 2009.
- [Shah *et al.*, 2018] Dev Shah, Haruna Isah, and Farhana Zulkernine. Predicting the effects of news sentiments on the stock market. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4705–4708. IEEE, 2018.
- [Tang *et al.*, 2018] Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. Why self-attention? a targeted evaluation of neural machine translation architectures. *arXiv preprint arXiv:1808.08946*, 2018.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [Wu *et al.*, 2018] Huizhe Wu, Wei Zhang, Weiwei Shen, and Jun Wang. Hybrid deep sequential modeling for social text-driven stock prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1627–1630. ACM, 2018.
- [Xi *et al.*, 2017] Edgar Xi, Selina Bing, and Yang Jin. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*, 2017.
- [Xie *et al.*, 2013] Boyi Xie, Rebecca Passonneau, Leon Wu, and Germán G Creamer. Semantic frames to predict stock price movement. In *Proceedings of the 51st annual meeting of the association for computational linguistics*, pages 873–883, 2013.
- [Xu and Cohen, 2018] Yumo Xu and Shay B Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1970–1979, 2018.
- [Yates *et al.*, 2007] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. Texrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics, 2007.
- [Zhang *et al.*, 2018] Suofei Zhang, Wei Zhao, Xiaofu Wu, and Quan Zhou. Fast dynamic routing based on weighted kernel density estimation. *arXiv preprint arXiv:1805.10807*, 2018.
- [Zhao *et al.*, 2018] Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*, 2018.

The FinSBD-2019 Shared Task: Sentence boundary detection in PDF Noisy text in the Financial Domain

Abderrahim Ait Azzi¹, Houda Bouamor², Sira Ferradans¹

¹Fortia Financial Solutions, France

²Carnegie Mellon University in Qatar, Qatar
name.surname@fortia.fr, hbouamor@cmu.edu

Abstract

In this paper, we present the results and findings of The FinSBD-2019 Shared Task on Sentence boundary detection in PDF Noisy text in the Financial Domain. This shared task was organized as part of The First Workshop on Financial Technology and Natural Language Processing (FinNLP), collocated with IJCAI-2019. The shared task aimed at collecting systems for extracting well segmented sentences from Financial prospectuses by detecting and marking their beginning and ending boundaries. The FinSBD shared task is the first to target the task of sentence boundary detection in the domain of Finance. A total of 9 teams from 7 countries participated in the shared task with a variety of systems and techniques.

1 Introduction

A vast amount of documents are constantly published online in machine-readable formats (generally PDF), containing not only text, but also other elements such as tables, images, and graphics. Therefore, most of the established PDF to text conversion products on the market (i.e., pdf2text) generate highly noisy unstructured texts containing abbreviations, non-standard words, false starts, missing punctuation, missing letter case information, and other text disfluencies. Building NLP applications customized for such texts is very challenging as most of the NLP tools (i.e. POS tagging, parsing, etc.) and applications (i.e. information extraction, machine translation) require as input a well-formatted clean text, where sentence boundaries are clearly marked [1].

Despite its important role in NLP, sentence boundary detection (SBD) has so far not received enough attention. Previous research in the area has been confined to formal texts only (news, European Parliament proceedings, etc.) where existing rule-based and machine learning approaches are extremely accurate (when the data is perfectly clean). No sentence boundary detection research to date has addressed the problem in noisy texts extracted automatically from machine-readable formats (generally PDF file format) files such as financial documents.

In this shared task, we focus on extracting well segmented sentences from Financial prospectuses by detecting and marking their beginning and ending boundaries. These are official PDF documents in which investment funds precisely describe their characteristics and investment modalities. The most important step of extracting any information from these files is to parse them to get noisy unstructured text, clean it, format information (by adding several tags) and finally, transform it into semi-structured text, where sentence boundaries are well marked.

In this paper we report the results and findings of the FinSBD-2019 shared task.¹ The Shared Task was organized as part of The First Workshop on Financial Technology and Natural Language Processing (FinNLP), collocated with IJCAI-2019.²

A total of 9 teams from 7 countries submitted runs and contributed 7 system description papers. All system description papers are included in the FinNLP workshop proceedings and cited in this report.

The large number of teams and submitted systems suggests that such shared tasks can indeed generate significant interest in the Finance and NLP research community.

2 Previous Work on SBD

While SBD is a foundational pre-processing task, previous research has been confined to clean texts in standard areas such as the news and limited datasets such as the WSJ corpus [2] or the Brown corpus [3]. SBD has been largely explored following several approaches that could be classified into three major classes: (a) rule-based SBD, using hand-crafted heuristics and lists [4]; (b) machine learning approaches to SBD [5; 6; 7; 3]; and more recently (c) deep learning methods [8]. Most of these approaches give fairly accurate results. These systems are based on a number of assumptions [4] that do not hold for noisy texts extracted automatically from PDFs (data is perfectly clean).

¹<https://sites.google.com/nlg.csie.ntu.edu.tw/finnlp/shared-task-finsbd>

²<https://sites.google.com/nlg.csie.ntu.edu.tw/finnlp/home>

```
{'text': " UFF Sélection Alpha AINFORMATIONS CLÉS POUR L' INVESTISSEUR
  Ce document fournit des informations essentielles aux investisseurs de cet OPCVM .
  Il ne s' agit pas d' un document promotionnel . Les informations qu ' il contient vous
  sont fournies conformément à une obligation légale , afin de vous aider à comprendre
  en quoi consiste un investissement dans ce fonds et quels risques y sont associés . ..." ,
'begin_sentence': [8, 21, 31 , ...],
'end_sentence': [20, 30, 66, ...] }
```

Figure 1: Example of the data json file

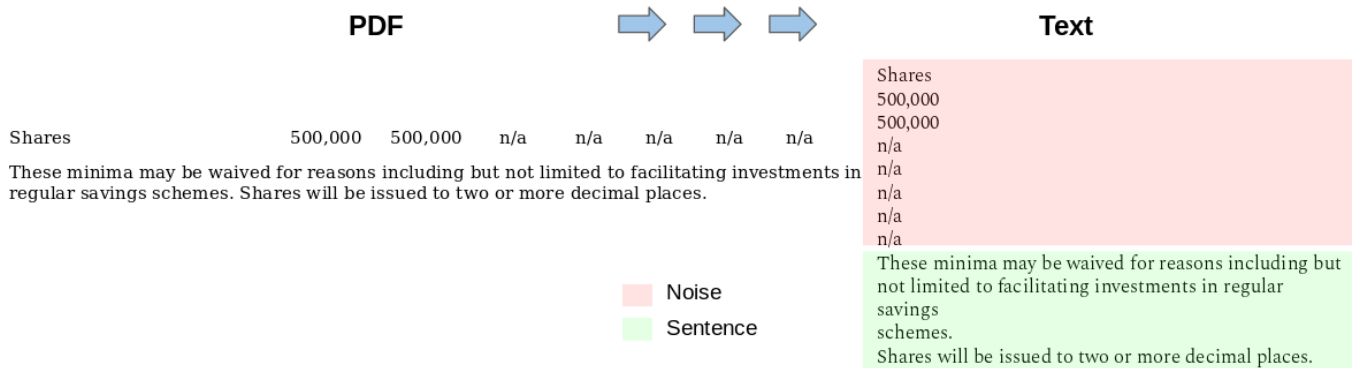


Figure 2: Example of a pdf to text conversion: Text extracted from a prospectus in a PDF format, spanning on several lines and with no punctuation marks (Target sentences are highlighted in green, Noise in Red)

Read et al., [1] conducted an analysis and review of commonly used *SBD* tools, but with a focus on generalization towards user-generated Web content. They evaluated several systems on a variety of data sets and report a performance decrease when moving from corpora with formal language to those that are less formal. Thus, designing and implementing approaches customized to different domains attracted the attention of several researchers. Griffis et al., [9] evaluated popular off-the-shelf NLP toolkits on the task of *SBD* for a set of corpora in the clinical domain. López and Pardo [10] tackle *SBD* on informal user-generated content such as web reviews, comments, and posts. Rudrapal et al., [11] present a study on *SBD* in social media context. *SBD* from speech transcriptions has also gained a lot of attention due to the necessity of finding sentential segments in the stream of transcripts, automatically recognized [12; 13].

Although text extracted from financial documents such as financial prospectuses, faces problems of text quality caused by segmentation issues, *SBD* (similarly to most other NLP tasks) has not received much attention in this domain.

3 Task Description

As part of the First Workshop on FinTech and Natural Language Processing (FinNLP), we introduced the FinSBD shared task which aims at sentence boundary detection in noisy text extracted from financial prospectuses, in two languages: English and French. Systems participating in this shared task were given a set of textual documents extracted from pdf files, which are to be automatically segmented to extract a set of well delimited sentences (clean sentences). The data will be in a json format (i.e. figure 1) containing: "**text**", that corresponds to the text to segment, "**begin_sentence**" and "**end_sentence**" correspond to all indexes of tokens marking the beginning and the end of well formed sentences in the text. It is important to note that the provided text is already segmented at the word level. All participants were asked to keep this segmentation since all tokens indexes are built based on it. The first token in the text has then the index 0 .

As stated in section 2, most of the previous research on sentence segmentation has been confined to clean texts in standard areas such as the news and limited datasets such as the WSJ corpus. However, the task of segmenting sentences extracted from noisy text, and more specifically text resulting from pdf conversion in the domain of finance is not much explored in the literature.

Figure 2 illustrates an example of a text extracted automatically from an English financial prospectus containing numerous issues ranging from missing punctuation to sentences spanning on several lines, in addition to the non-standard capitalization (very typical in financial texts).

Other issues are caused by the ambiguous use of full stop punctuation marks in several section numbers (i.e.,

"1.", "2.") and to mark the end of a sentence. Also, the dash sign (-) could be used as a hyphen or to mark the math minus sign. Moreover, financial prospectuses contain a large number of financial institutions names that appear with their legal form abbreviations (i.e., "S.A" for Société Anonyme, "LTD." for Limited Company, etc. Hence, applying commonly used sentence segmentation tools (i.e., Stanford sentence segmenter [14]) that typically rely on punctuation marks or capitalization in the sentence boundary detection (*SBD*) process is impractical.³

4 Shared Task Data

Next, we discuss the corpora used for the English and French subtasks.

4.1 Corpus annotation

Financial prospectuses are available online in a pdf format and are also made available from asset managers. We compiled a list of 11 prospectuses in English (140 pages on average) and 92 in French (25 pages on average). These prospectuses are first converted to a text document format using the freely available tool pdf2text⁴. Every line in these documents is tokenized at the word level. We extend the Keras tokenizer by adding several rule-based functions to take into account more cases (i.e., possessive form of words, acronym detection, etc.). We remove all non-ASCII characters resulting from the conversion step except the French accents.

We provided three bi-lingual (English and French) annotators with text files in both languages extracted automatically from financial prospectuses, along with their original PDFs.⁵ We gave them detailed annotation guidelines and asked them to go through every text segment, understand it and mark the boundaries of what they estimate corresponds to a sentence. A sentence is defined as a set of words that is complete in itself, typically containing a subject and predicate, conveying a statement, question, exclamation, or command, and consisting of a main clause and sometimes one or more subordinate clauses. We deliberately asked them not to rely on capitalization and punctuation markers only. We use the online annotation framework BRAT [15]. The tool displays to the annotator each document in a text segment per line format.

The annotation labels used in building the corpus are the following:

- **Begin_Sentence (BS)**: marks the first token of a sentence which can be a word, a character or bullet, etc.
- **End_sentence (ES)**: denotes the token that comes at the end of a sentence whether it is a punctuation mark or a word in case the sentence does not end with a punctuation.

³<https://nlp.stanford.edu/software/tokenizer.html>

⁴<http://www.pdf2text.com/>

⁵This helps visualize the text in its original clean setup, which helps the annotators locate sentences more rapidly.

Annotation Challenges . The data annotation process was an arduous task due to:

1. the absence of the layout creates many ambiguities: some headers are transformed to lower-cased words and are put in the same lines as the sentences. For sentences that begin at the end of a PDF page and end in the next one, the footer content enters inside such sentences in the text file which makes it impossible to annotate them. So in this case, the solution was to remove manually these footers.
2. the absence of end of sentence punctuation markers. In fact, some sentences do not end with full stops e.g. "The DJ - UBS Index generally rolls the futures contract which is closest to expiry into the futures"
3. the punctuation errors appearing in some sentences, such as a period in the middle of a sentence e.g. *This Supplement forms part of the Prospectus dated 1 January 2015 for GAM Star (Lux) SICAV . and should be read in conjunction with that Prospectus.*
4. the excessive use of in-sentence lists mainly in English prospectuses. An in-sentence list is a sentence that contains a list of ordered sub-sentences usually using letter (a), (b) and so on, or numbers (1), (2) and so on.
5. the excessive use of Uppercase words that are neither proper nouns nor named entities (Shares, Class, Initial Subscription,..) which makes it less obvious to spot the beginning of a sentence.

4.2 Corpus Description

In the following, we provide an analysis of the data used for both subtasks: English and French.

In Table 1, #prospectuses indicates the number of prospectuses that were used in each data set; #Types is the total number of unique tokens in the text; and #Sentences is the number of segmented sentences in the text. % *OOV* words represents the rate of Out-of-Vocabulary words. We notice that the French Validation/Testing data contain higher *OOV* rate the English data ($\approx +5\%$).

In order to extend our analysis, we first report the percentage of sentences ending with a punctuation mark such as the full stop, column, and semi-column. Although this rate is higher than 93% for both language, it still shows that there are many sentences that do not contain any ending indicator as mentioned in the previous section notably in the french testing set ($\approx 7\%$). Then, we report the percentage of sentences that start with a capital letter. This percentage is around 85% for the English data, which means that in many cases, capitalization is not an indicator of the beginning of sentences, which shows that our task is more sophisticated than the traditional SBD tasks.

5 Participants and Systems

A total of 69 teams registered in the shared task, out of which 8 submitted a paper with the description of their method. The participants came from 7 different

countries and belonged to 10 different institutions. The shared task was a success in bringing together private and public research institutions. As private, Accenture AI Labs, SeerNet Technologies LLC, OPT inc and AIG. As public, Heidelberg Institute for theoretical studies, University of Kyoto, Insight Center for Data Analytics (National University of Ireland Galway), the Hong Kong Polytechnic University and Harbin Institute of Technology (see Table 3 for more details).

In table 2, we show the details on the submissions per task. It is important to note that not all the participants that submitted a standard run, sent a paper describing their approach.

Participating teams explored and implemented a wide variety of techniques and features. In this section, we give a brief description of each system, more details could be found in the description papers appearing in the proceedings of the FinNLP 2019 Workshop.

Most participants formulated the problem as either a sequence-labeling task or as a word level classification task. In this context, the best performing methods are those that used word embeddings with a neural model mainly based on LSTMs, although other features were explored. Below is a short summary of each participating system. Teams SeerNet and ISI do not appear because they did not send a descriptive paper.

AI_blues [19] In this work, the problem is stated as a sequence labeling task for which the participants use a CRF method. The input features for the CRF are mostly defined based on punctuation, lexical combinations of numbers and letters, presence of upper case letters, POS tags and some basic features (token length, is upper case, is lower case, token type).

NUIG [18] This team was the only one that took into account and explored the financial component of the task. They trained several character-level RNN embeddings using external financial text. These embeddings were used together with pretrained GLOVE [23] (for English) and FastText [24] (for French) embeddings. Finally, the system performs a sequence labeling using a BiLSTM-CRF model.

PolyU [22] The team defines a set of handcrafted features such as punctuation, presence of upper case, acronyms, and POS tags and train two models: (1) a multilayer neural network and (2) a random forest model. They show that cross-lingual training improves results for both languages.

mhirano [20] The features used in this system are pre-trained word2vec embeddings, POS tags, presence of capital letters, and alpha-numerical patterns. They serve as input to a multilayer perceptron trained to classify the central word of a given window. They also propose a second method that is rule-based and defined on sequences of token types.

HITS-SBD [21] This team proposed two methods: (1) random forest classifier on top of a TF-IDF representation

	English			French		
	Training	Validation	Testing	Training	Validation	Testing
# Prospectuses	9	1	1	74	9	9
# Tokens	904057	49859	56952	827852	119008	106577
# Types	13478	2926	3651	14267	6267	5610
# Sentences	22342	1384	1265	22636	3141	2981
% <i>OOV</i> words	—	10.93	13.75	—	15.9	19.22
% Punct. as end sentence	93.31	97.97	97.0	94.39	96.18	93.62
% Uppercase begin sentence	87.33	82.80	84.35	89.90	90.06	89.23

Table 1: Distribution of the Training, Validation and Testing sets used in the English and French corpora.

	# teams	# std runs
subtask EN	9	18
subtask FR	7	15
papers	8	-

Table 2: Statistics on the participation in the French and English subtasks.

of the word context, and (2) a ruled-based method based on pattern matching.

aiai [17] They defined the task as a classification task of the center word of a given window. They use two classification methods: (1) LSTM with attention and (2) CNN, both on top of pretrained Glove word level embeddings and specific word embeddings that encode upper case letters.

AIG [16] Similarly to many of the other teams, AIG implemented two models: (1) BI-LSTM with CRF on top of pretrained GLOVE word embeddings, and (2) a fined-tuned version of BERT for the sequence labeling.

6 Results and Discussion

In this section, we describe the evaluation metrics used in the shared task and we give an analysis of the results obtained for the various submitted systems.

Evaluation Metric Participating systems are ranked based on the macro-averaged F1 scores obtained on blind test sets (official metric). We also report the scores of **Begin_Sentence** (BS) and **End_Sentence** (ES), that are computed separately.

Table 4 reports the results obtained on FinSBD English by the teams detailed in the previous section. For the results on FinSBD French, please check table 5.

Team	English		
	BS	ES	Average
AIG1	0.88	0.89	0.885
seernet1	0.85	0.9	0.875
aiail	0.83	0.91	0.87
isi1	0.83	0.89	0.86
NUIG1	0.81	0.9	0.855
isi2	0.82	0.89	0.855
AIG2	0.83	0.88	0.855
AI_Blues2	0.82	0.87	0.845
AI_Blues1	0.82	0.87	0.845
mhirano1	0.78	0.89	0.835
aiai2	0.79	0.88	0.835
NUIG2	0.81	0.85	0.83
HITS-SBD2	0.8	0.86	0.83
HITS-SBD1	0.8	0.86	0.83
PolyU_CBS-CFA_NN1	0.77	0.86	0.815
PolyU_CBS-CFA_RFC1	0.7	0.86	0.78
PolyU_CBS-CFA_RFC2	0.68	0.86	0.77
mhirano2	0.58	0.67	0.625

Table 4: Results obtained by the participants for the FinSBD English task. The teams are ordered by the F1 average value (last column).

Team	French		
	BS	ES	Average
seernet	0.91	0.93	0.92
aiail	0.91	0.92	0.915
NUIG1	0.9	0.92	0.91
NUIG2	0.9	0.92	0.91
isi1	0.9	0.91	0.905
isi2	0.89	0.91	0.9
AI_Blues1	0.85	0.88	0.865
AI_Blues2	0.84	0.88	0.86
PolyU_CBS-CFA_RFC1	0.84	0.88	0.86
mhirano1	0.82	0.89	0.855
PolyU2	0.83	0.87	0.85
PolyU_CBS-CFA_NN1	0.83	0.87	0.85
PolyU_CBS-CFA_RFC2	0.81	0.88	0.845
mhirano2	0.67	0.68	0.675
aiai2	0.01	0.02	0.015

Table 5: Results obtained by the participants for the FinSBD French task. The teams are ordered by the F1 average value (last column).

Discussion Simple ruled-based methods based on obvious punctuation characters can perform very well on SBD, but in order to perform extremely well, we need to take into account the long tail exceptions specially

Team	Affiliation	Tasks
AIG [16]	American International Group, United Kingdom	English only
seernet	SeerNet Technologies, LLC, India	English and French
aiai [17]	OPT, Inc and Herbin institute of technology, Japan and China	English and French
isi	Information Sciences Institute (University of Southern California), USA	English and French
NUIG [18]	National University of Ireland Galway, Ireland	English and French
AI_Blues [19]	Accenture Solutions Pvt Ltd, India	English and French
mhirano [20]	The University of Tokyo, Japan	English and French
HITS-SBD [21]	Heidelberg Institute for Theoretical Studies, Germany	English only
PolyU_CBS [22]	The Hong Kong Polytechnic University, China	English and French

Table 3: List of the 9 teams that participated in Subtasks English and French of the FinSBD Shared Task.

present in noisy financial text extracted from pdf, which is the target corpus of this shared task. We can see this in the results. Two ruled-based methods, mhirano2 and HITS-SBD2, were proposed obtaining the 18th (0.625 F1 score in EN) and 13th position (0.83 F1) respectively. All the other methods implemented machine learning algorithms (AI_blues, HITS-SBD1, PolyU2) and deep learning methods (NUIG, PolyU1, mhirano1, aig, aiain). The best performing teams (NUIG1, aig1 and aiain1) implemented similar models: on top of GLOVE word embeddings a combination of (bi-)lstm with a CRF or attention layer.

Very little attention was paid to the fact that the corpus was from the financial domain. Only one team used financial features by training a language model on external financial text.

Finally, most participants understood how similar the task was to POS tagging and either used POS tags as features (PolyU, mhirano AI_Blues) or took inspiration from learning methods that performed well in POS tagging tasks (NUIG1).

7 Conclusions

In this paper we presented the setup and results for the FinSBD-2019 Shared Task on Sentence boundary detection in PDF Noisy text in the Financial Domain, organized as part of The First Workshop on Financial Technology and Natural Language Processing (FinNLP), collocated with IJCAI-2019. A total of 69 people registered and 9 teams from 7 countries participated in the shared task with a wide variety of techniques. The most successful methods were based on word embeddings (mostly GLOVE) followed by a (bi)lstm-crf (or an attention mechanism). The best average F1 score on the FinSBD French task was 0.92 and 0.885 for the FinSBD English.

We introduced a new data set on the SBD problem in text automatically extracted from PDF files for French and English. This scenario is very realistic in everyday applications which may explain the diversity of institutions that participated, from public universities to for profit organizations from the financial domain. In this sense, the shared task was a success since it was able to

bring together researchers from different sectors.

Acknowledgments

We would like to thank our dedicated annotators who contributed to the building the French and English corpora used in this Shared Task: Anais Koptient, Aouataf Djillani, and Lidia Duarte.

References

- [1] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India, December 2012. The COLING 2012 Organizing Committee.
- [2] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.
- [3] Dan Gillick. Sentence boundary detection and the problem with the us. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244. Association for Computational Linguistics, 2009.
- [4] Gregory Grefenstette and Pasi Tapanainen. What is a word, what is a sentence?: problems of tokenisation. 1994.
- [5] Michael D Riley. Some applications of tree-based modelling to speech and language. In *Proceedings of the workshop on Speech and Natural Language*, pages 339–352. Association for Computational Linguistics, 1989.
- [6] Jeffrey C Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, pages 16–19. Association for Computational Linguistics, 1997.

- [7] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.
- [8] Marcos V Treviso, Christopher D Shulby, and Sandra M Aluisio. Evaluating word embeddings for sentence boundary detection in speech transcripts. *arXiv preprint arXiv:1708.04704*, 2017.
- [9] Denis Griffis, Chaitanya Shivade, Eric Fosler-Lussier, and Albert M Lai. A quantitative and qualitative evaluation of sentence boundary detection for the clinical domain. *AMIA Summits on Translational Science Proceedings*, 2016:88, 2016.
- [10] Roque López and Thiago AS Pardo. Experiments on sentence boundary detection in user-generated web content. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 227–237. Springer, 2015.
- [11] Dwijen Rudrapal, Anupam Jamatia, Kunal Chakma, Amitava Das, and Björn Gambäck. Sentence boundary detection for social media text. In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 254–260, 2015.
- [12] González-Gallardo, Carlos-Emiliano, Torres-Moreno, and Juan-Manuel. Sentence boundary detection for french with subword-level information vectors and convolutional neural networks. *arXiv preprint arXiv:1802.04559*, 2018.
- [13] Chenglin Xu, Lei Xie, and Xiong Xiao. A bidirectional lstm approach with word embeddings for sentence boundary detection. *Journal of Signal Processing Systems*, pages 1–13, 2017.
- [14] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.
- [15] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, Avignon, France, April 2012. Association for Computational Linguistics.
- [16] Yan Huang Jinhua Du and Karo Moilanen. Aig at the finsbd task: Sentence boundary detection through sequence labelling and bert fine-tuning. In *The First Workshop on Financial Technology and Natural Language Processing of IJCAI 2019*, 2019.
- [17] Ke Tian and Zi Jun Peng. aiai at finsbd task: Sentence boundary detection in noisy texts from financial documents using deep attention model. In *The First Workshop on Financial Technology and Natural Language Processing of IJCAI 2019*, 2019.
- [18] Tobias Daudert and Sina Ahmadi. Nuig at the finsbd task: Sentence boundary detection for noisy financial pdfs in english and french. In *The First Workshop on Financial Technology and Natural Language Processing of IJCAI 2019*, 2019.
- [19] Ditty Mathew and Chinnappa Guggilla. Ai_blues at finsbd shared task: Crf-based sentence boundary detection in pdf noisy text in the financial domain. In *The First Workshop on Financial Technology and Natural Language Processing of IJCAI 2019*, 2019.
- [20] Kiyoshi Izumi Masanori Hirano, Hiroki Sakaji and Hiroyasu Matsushima. mhirano at the finsbd task: Pointwise prediction based on multi-layer perceptron for sentence boundary detection. In *The Workshop on Financial Technology and Natural Language Processing of IJCAI 2019*, 2019.
- [21] Mehwish Fatima and Mark-Christoph Mueller. Hitsbd at the finsbd task: Machine learning vs. rule-based sentence boundary detection. In *The First Workshop on Financial Technology and Natural Language Processing of IJCAI 2019*, 2019.
- [22] Emmanuele Chersoni Natalia Klyueva Kathleen Ahrens Bin Miao David Broadstock Jian Kang Amos Yung Mingyu Wan, Rong Xiang and Chu-Ren Huang. Sentence boundary detection of financial data with domain knowledge enhancement and cross-lingual training. In *The First Workshop on Financial Technology and Natural Language Processing of IJCAI 2019*, 2019.
- [23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.
- [24] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

AIG Investments.AI at the FinSBD Task: Sentence Boundary Detection through Sequence Labelling and BERT Fine-tuning

Jinhua Du, Yan Huang and Karo Moilanen

Investments AI, AIG (American International Group, Inc.)

{ Jinhua.Du | Yan.Huang | Karo.Moilanen }@aig.com

Abstract

This paper describes the method that Investments AI at AIG (American International Group, Inc.) submitted to the FinSBD-2019 shared task (“*Sentence Boundary Detection (SBD) in PDF Noisy Text of the Financial Domain*”) to extract meaningful, well-formed sentences from noisy unstructured financial text. We approach sentence boundary detection as a sequence labelling task to recognise the start and end token boundaries of sentence(-like) constructs. We evaluated two neural architectures, namely 1) Bidirectional Long Short-Term Memory (BiLSTM) and 2) Bidirectional Encoder Representations from Transformers (BERT). Our extensive experiments on the official FinSBD-2019 datasets demonstrate that a fine-tuned BERT model with customised hyper-parameters (BERT-SBD) outperforms BiLSTM models in several evaluation metrics. Our BERT-SBD submission ranked first on the English test set in terms of *MEAN F1* score in the joint sentence-begin-and-end test condition.

1 Introduction

The sentence is one of the most prominent building blocks in practical NLP and formal linguistics alike. Many, ultimately leaky, definitions for what a sentence is (not) can be found in both communities. At the level of informal common sense, a sentence is taken to represent a “complete thought”¹. In Halliday’s functional-thematic interpretation, the sentence is a basic unit of information composed of a topic (theme) and a comment; and the highest graphological unit of punctuation which conventionally begins with an upper-case letter and ends with a full stop [Halliday, 2004]. The sentence is conventionally the structurally highest construct in formal syntax (lexicogrammar), typically a clause complex or minimally at least one main clause (a predicator with an internal subject complement) [Huddleston and Pullum, 2002]. Many NLP applications and data sets² view sentences as arbitrarily

¹For example, academic writing guides (<https://www.uts.edu.au/sites/default/files/article/downloads/sentence.pdf>)

²For example, the “*Brief one-sentence movie summary*” field in <https://www.kaggle.com/PromptCloudHQ/imdb-data>

truncated text snippets which are simply ‘useful’ in practical terms.

Regardless of how the sentence is defined formally, sentence boundary detection (SBD) (cf. sentence boundary disambiguation, sentence segmentation, sentence breaking, sentence chunking) is a foundational, critically important upstream step in many NLP applications and (sub)tasks, such as part-of-speech tagging, named entity recognition, dependency parsing, and semantic role labelling, to name a few. Sentence boundary detection attempts to determine the spans (bounds, begin/from-end/to token indices) of sentences and sentence-like constructs below paragraphs, sections, or other suprasentential structures. Because incorrect sentence spans can propagate and generate noise (and undesirable complications) for downstream tasks, SBD plays a critical role in practical NLP applications.

Despite its importance, SBD has received much less attention in the last few decades than some of the more popular subtasks and topics in NLP. On the one hand, (superficially) high baseline performance levels can be achieved by naive lookup methods that capture obvious, frequent sentence-final punctuation characters such as [!?”] in conjunction with elementary space and case heuristics [Reynar and Ratnaparkhi, 1997]. Such baselines leave little room for further optimisation on traditional test sets derived from formal news(wire) sources. On the other hand, the long tail of exceptions in SBD makes the task non-trivial and challenging: a good majority of potential sentence boundary markers exhibit graphemic (and deeper semantic) ambiguity, particularly the full stop (period) which occurs in abbreviations, initials, honorifics, ordinal numbers, email addresses, ellipses, and the like [Stamatatos and Fakotakis, 1999; Kiss and Strunk, 2006; Gillick, 2009].

Beyond traditional, well-formed, -edited, and -curated news data, the snowballing of noisy web and social media data since the late 1990s has made SBD much harder: when faced with unstructured user-generated content involving tweets, extremely complex graphemic devices (e.g. new emoji, abbreviations, and acronyms), mark-up, and (up to a point) machine-readable data, traditional (and most off-the-shelf) sentence breakers that were trained on “bare” ASCII data in the Penn Treebank (PTB) simply run out of steam [Gimpel *et al.*, 2011; Read *et al.*, 2012]. Canonical SBD approaches optimised for the canonical news genre en-

counter many complications even in other formal domains such as the biomedical [Griffis *et al.*, 2016] or legal [Savelka *et al.*, 2017] ones.

Financial documents which are replete with extremely complex sentences provide one of the most unforgiving but also rewarding application domains for any SBD method. Addressing the lack of SBD research in financial NLP, the FinSBD-2019 shared task [Ait Azzi *et al.*, 2019] focused on the specific challenges that come with noisy financial texts, including impure data extracted and converted automatically from machine-readable formats (such as PDFs). The main task was to detect the spans (begin(ning)/from vs. end(ing)/to token boundaries) of “well-formed sentences” in financial prospectuses - official PDF documents³ published by investment funds to describe their products to clients.

Datasets for the shared task were released in two languages, English and French. We participated only in the English track. Our system, which relies on state-of-the-art neural models and fine-tuning techniques, approached the FinSBD-2019 challenge as a generic sequence labelling task.

According to the organisers’ automatic evaluation metrics, we reached the highest *MEAN F1* score in the English subtask with a 1-point margin over the second-best submission.

2 Task Definition

The majority of past approaches to sentence boundary detection fall into three broad classes: (a) rule-based methods which typically exploit hand-crafted character and spacing heuristics, lookup patterns (e.g. Stanford CoreNLP⁴), or syntactic dependencies (SpaCy⁵); (b) supervised machine learning trained on sentence boundary annotations; and (c) unsupervised machine learning with raw, unlabelled corpora ([Read *et al.*, 2012]).

In practical NLP work, rule-based methods are still popular as they offer the quickest and cheapest way to achieve reasonable performance levels for many NLP tasks. However, if labelled boundary annotations are available, supervised machine learning methods tend to offer greater recall. Previous supervised methods use various strategies to define SBD as a form of classification, for example (i) binary classification to classify each occurrence of `[.!?]` as a valid vs. invalid sentence boundary marker [Reynar and Ratnaparkhi, 1997], or (ii) sequence labelling over multiple classes to tag each token (commonly using a BIO (IOB) tagging scheme [Evang *et al.*, 2013]).

As FinSBD-2019 provided training data with boundary labels (beginning vs. ending) for each token in text, we opted for classification and evaluated state-of-the-art supervised neural models to classify each token in the text to a given class in conjunction with sequence labelling.

We observed the following in the training and development sets of FinSBD-2019 data:

- We found **953** distinct beginning tokens, where determiners, prepositions, conjunctions, and particles such

³Sample prospectus: <https://bit.ly/2QztxR0> (via Google).

⁴<https://stanfordnlp.github.io/CoreNLP/ssplit.html> [Manning *et al.*, 2014]

⁵<https://spacy.io/api/annotation#sentence-boundary>

as *A, The, In, For, And* cover more than 50% compared to nouns, pronouns, digits, and miscellaneous single-character constructs (e.g. *Investments, LUXEMBOURG, a, b, I, 2*).

- We found **207** distinct ending tokens, where the full stop, semicolon, and colon cover more than 90% compared to ordinary nouns, numerical tokens (e.g. (year “2014”), and the like. Note that most traditional sentence boundary gold standards do not use such implicit, structurally opaque tokens as sentence boundary markers.

Regarding well-formedness, we observe that the majority of FinSBD-2019 annotations appear to capture sentence or sentence-like constructs which fall under conventional definitions (cf. Section 1). However, many exceptions can be found in the data set, for example constructs devoid of a main verb or a sentence-final period, and other largely arbitrary fragments. Some example sentence(-like) annotations from the training data are shown below.

*All Shares will be issued in registered form .
(b) the legal requirements and
Any member state of the EU .
– bonds and other forms of securitised debt ,*

We are not aware of any inter-annotator agreement scores that would estimate human performance in financial sentence boundary detection, and shed light on the quality and reliability of FinSBD-2019 sentence annotations.

Owing to the fact that 1) both sentence beginning and ending tokens need to be recognised, and 2) punctuation characters do not alone suffice as ending tokens, we define the following three (3) classes for each token to be used in sequence labelling

- S: Start (sentence-initial token)
- E: End (sentence-final token)
- O: Other (sentence-internal token, neither the start nor the end)

which we use to annotate tokens in sentences such as

The/S company/O made/O £10k/O during/O 2015/O ./E

3 SBD Systems

Deep neural networks (DNN) have pushed the state of the art in many areas of NLP. A DNN model learns a hierarchy of nonlinear feature detectors that can capture more and more complex syntactic and semantic representations. Two DNN architectures are particularly popular, namely recurrent neural networks (RNN) with long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] cells or gated recurrent units (GRU) [Cho *et al.*, 2014], and Transformer [Vaswani *et al.*, 2017] which exploits feedforward neural networks and multi-head self-attention mechanisms. We chose two open source systems that are variants of these two architectures for our submission, namely BiLSTM-CRF [Ma and Hovy, 2016] and BERT [Devlin *et al.*, 2018].

3.1 BiLSTM-CRF

An RNN or LSTM [Hochreiter and Schmidhuber, 1997] maintains a memory based on a history which enables the model to predict the current output on the basis of past information and outputs. Bidirectional LSTM [Schuster and Paliwal, 1997] is a variant of unidirectional LSTM which connects two hidden layers of opposite directions to the same output so it can capture information from past and future states simultaneously. In a sequence labelling task, we can efficiently access both past (via forward states) and future (via backward states) input representations for a specific time step t , as shown in Fig. 1.

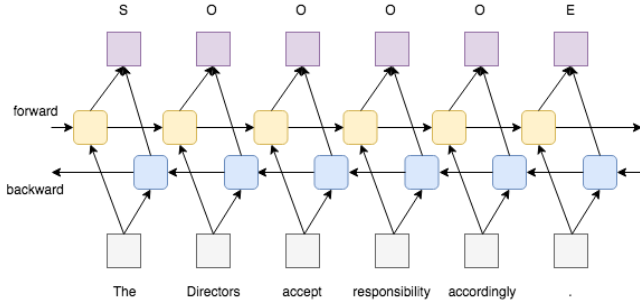


Figure 1: A BiLSTM architecture for sequence labelling for SBD.

We can see that each input token is encoded to a hidden state by the forward and backward LSTM network, respectively, through the integration of previous context information. In the output layer, two hidden states from the forward and backward networks are typically concatenated and then fed into a softmax function to generate a probability distribution for a given label set. The label with the highest probability is conventionally chosen as the final prediction.

Although the current hidden state in an LSTM network does exploit a limited history, the previous neighbour tag is not used when the current tag in the final output layer is predicted. However, the linear order of tags does matter in many sequence labelling tasks. For example, in our SBD task, the sentence-initial start tag (S) has to precede the sentence-final end tag (E). To account for such constraints, the linear-chain Conditional Random Fields (CRF) model is often connected to the output layer of an LSTM network. Fig. 2 shows a hybrid BiLSTM-CRF architecture of this kind.

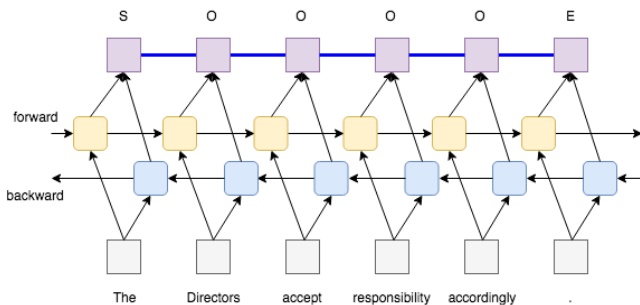


Figure 2: A BiLSTM-CRF architecture for sequence labelling for SBD.

In Fig. 2, the CRF network is represented by the blue lines which connect consecutive BiLSTM outputs. The CRF layer is parameterised by a state transition matrix which indicates the transition probability from one state to another. With such a layer, we can use past and future tags to predict the current tag, correspondingly.

The basic input unit for the BiLSTM-CRF network is conventionally a word (token) which is converted to a vector representation with a fixed dimension. Word vectors are generally pre-trained using neural networks on large-scale datasets (e.g. word2vec [Mikolov *et al.*, 2013], GloVe [Pennington *et al.*, 2014]). Pre-trained word embeddings can be used as initial values for input words or fine-tuned further during training. Pre-trained word embeddings, which can provide a boost for many NLP tasks, are convenient because task-specific training data sets tend to be relatively small. However, word-level inputs are not without their own complications the most prominent of which are 1) out-of-vocabulary (OOV) items, and 2) necessarily limited representative power regarding deeper semantics. Therefore, character-level embeddings are typically used in conjunction with word-level embeddings to represent words. Fig. 3 illustrates the use of a BiLSTM network to learn character-level embeddings for words.

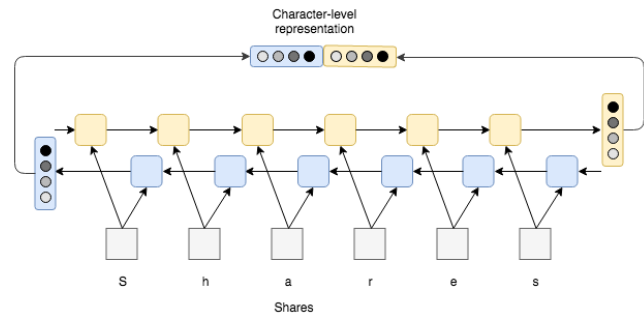


Figure 3: A BiLSTM network for character-level word representation.

The final input representation in our BiLSTM system is a concatenation of word-level embeddings derived from GloVe and character-level embeddings trained using the BiLSTM network. The complete architecture of our BiLSTM-CRF system for SBD is shown in Fig. 4.

The character representation in Fig. 4 is the output from the BiLSTM network for character-level word representation (see Fig. 3).

3.2 BERT

Conventional DNN models, which tend to require large datasets and which can take days to converge, are typically trained from scratch for a given task. Attention has recently moved towards more efficient transfer learning paradigms which first pre-train a DNN model on large datasets, and then fine-tune them towards a specific domain or task. Recent approaches have opted for pre-trained neural language models instead of pre-trained embeddings. BERT, which has achieved state-of-the-art performance in many NLP tasks, is the most representative pre-trained model in this regard.

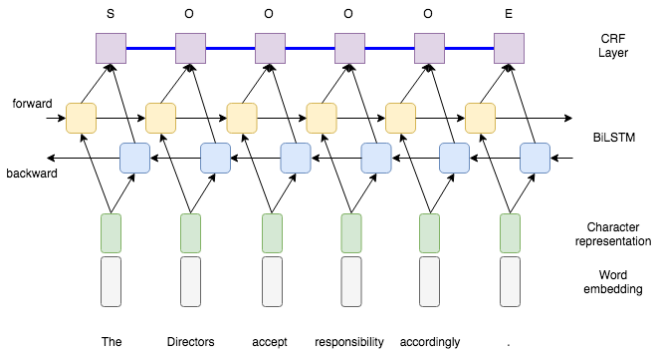


Figure 4: The BiLSTM-CRF architecture of our FinSBD-2019 submission.

BERT uses a multi-layer Transformer encoder [Vaswani *et al.*, 2017] to pre-train deep bidirectional representations by jointly conditioning on both left and right context across all layers [Devlin *et al.*, 2018]. As a result, pre-trained BERT representations can be fine-tuned conveniently using only one additional output layer. Fig. 5 illustrates the Transformer and BERT architectures.

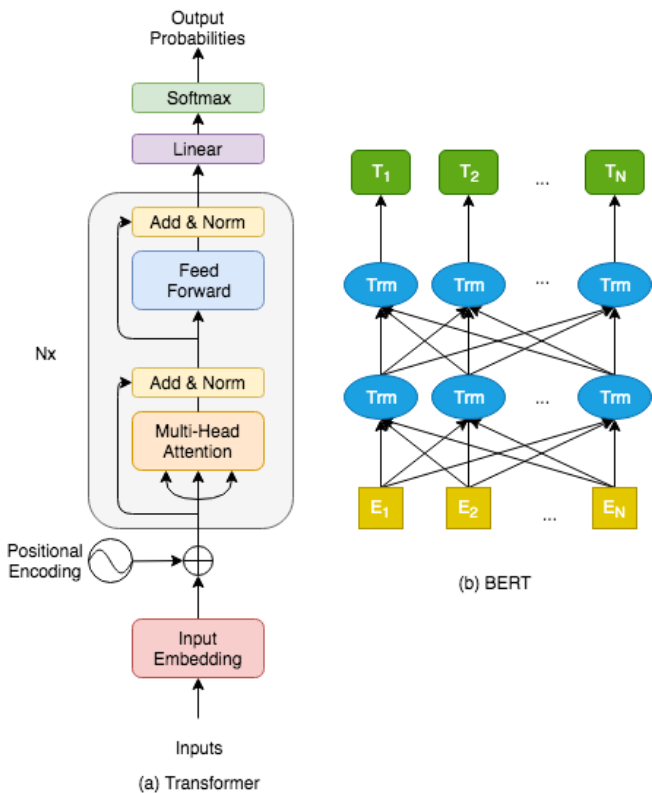


Figure 5: The Transformer (a) and BERT (b) architectures. NB. Trm refers to (a).

Transformer makes use of self-attention (instead of RNNs or CNNs) as its basic computational block. Transformer uses a combination of self-attention and feed-forward layers in the encoder. In the standard Transformer model, the en-

coder is composed of a stack of $N_x = 6$ identical layers, with each layer having two sublayers, namely a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. A residual connection is utilised around each sublayer, followed by layer normalisation.

An attention function can be described as a way to map a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key [Vaswani *et al.*, 2017].

For a given token, BERT’s input representation is constructed by summing the corresponding token, segment, and position embeddings. BERT is trained using two unsupervised prediction tasks, Masked Language Model and Next Sentence Prediction. To fine-tune BERT towards a sequence labelling task, the final hidden representation T_i for each token i is fed into a classification layer over the label set. The predictions are not conditioned on the surrounding predictions.

Since we view our SBD task as a sequence labelling problem, we configure BERT to instantiate the token tagging architecture shown in Fig. 6, where C is the hidden state for the first token in the input which corresponds to the special CLS word embedding.

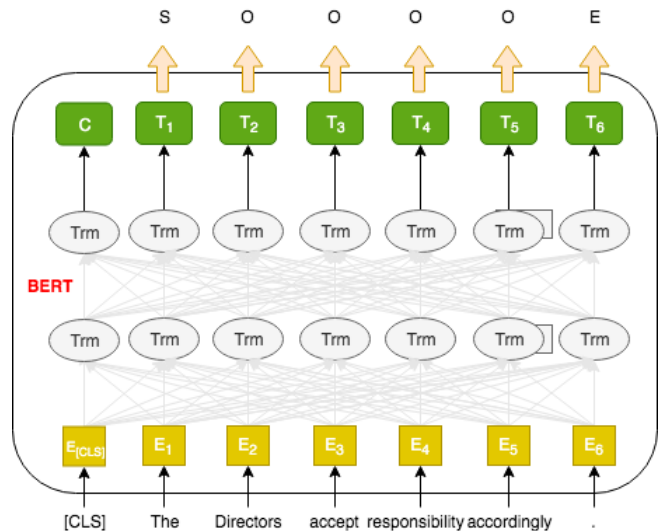


Figure 6: BERT fine-tuned towards sequence labelling for SBD.

4 Experiments

We built two neural systems using the official training data, and tuned their parameters on the validation set. This section summarises the data we used and the steps we took to build, fine-tune, test, and evaluate our systems.

4.1 Data

The FinSBD-2019 data set contains financial documents which had been pre-segmented automatically. The data

	Training	Validation	Testing
#Segment	57,497	2,036	2,505
#Token	904,057	49,859	56,952
#Vocabulary	12,047	2,843	3,539
#Sentence	22,342	1,384	1,265
Max Length	581	303	553
Min Length	1	1	1
Avg Length	15.7	24.5	22.7
Coverage (%)	–	89.69	86.89

Table 1: FinSBD-2019: summary statistics.

was provided as a JSON file which contains raw `text` (without any sentence boundaries) with accompanying `begin_sentence` and `end_sentence` token indices for sentence boundaries, respectively. The raw text had been pre-tokenised using NLTK [Loper and Bird, 2002]. The first token in the text is indexed 0. Table 1 shows summary statistics for the official FinSBD-2019 data set.

In Table 1, *#Segment* indicates the total number of segments (sequences of words on separate lines); *#Vocabulary* is the total number of unique tokens in the text; and *#Sentence* represents the total number of well-formed sentences in the text.

Note that the labels for the test set were released after submission. We can observe that the test set is somewhat different from the validation one, with more segments, tokens, and unique tokens, and fewer well-formed sentences in the former. This difference implies that the test set may be noisier or somehow more complicated, or simply of a poorer quality.

Max Length is the maximum length of the segment in the text, *Min Length* is the minimum segment length, and *Avg Length* is the average length of the segment over the text. It can be seen that the distribution of segment lengths is markedly unbalanced.

The *Coverage*, which indicates how many unique tokens from the validation or test set appear in the training set, can be used as a proxy to quantify the presence of out-of-vocabulary (OOV) or unknown tokens. We can see that the validation set and the test set are comparable in this regard.

4.2 Text Pre- and Post-Processing

We observe that the segments provided are not always correct syntactically, for example in cases where a (syntactic) sentence had been split across multiple segments. In such cases, we cannot use the provided segments as direct inputs to our SBD systems. We followed a simple text pre-processing strategy as follows:

- We remove all newline symbols in the text, and convert it to a single continuous token sequence.
- We split the resultant sequences into short(er) sequences through a sequence length parameter (L). We use $L = 60$ for our BiLSTM-CRF system and $L = 250$ for our BERT-SBD system in the submissions.
- We label each sequence using our pre-defined $\{S, E, O\}$ label set following the CoNLL2003 BIO (IOB) tagging scheme [Tjong Kim Sang and De Meulder, 2003].

Parameter	BiLSTM-CRF	BERT-SBD
Pre-trained model	–	bert-base-cased
Max seq length	60	256
Lower case	False	False
Batch size	20	32
Learning rate	0.001	5e-5
Learning decay	0.9	–
Train epochs	15	5
Dropout	0.5	–
Optimiser	Adam	–
Hidden size char	100	–
Hidden size SBD	300	–
Char embedding dim	100	–
Word embedding dim	300	–

Table 2: Parameter configurations for our submission.

- Tokens which are in an unsupported encoding or which cannot be recognised by BERT are tagged as *UNK*.

We also rely on an additional, simple post-processing strategy to process the outputs from the two SBD systems: for predictions with only one S or E, we look for simple punctuations and upper-case characters in limited context windows to reconstruct the missing E or S, correspondingly.

4.3 System Settings

The hyper-parameters are shown in Table 2. *Hidden size char* denotes the hidden size of BiLSTM for character-level embedding training while *Hidden size SBD* indicates the hidden size of BiLSTM-CRF for the SBD task. *bert-base-cased* has 12 layers with a hidden size of 768 and 12 multi-head attentions, with 110M parameters in total. BERT-SBD uses default configurations for other parameters.

We use WordPiece [Wu *et al.*, 2016] embeddings with a 30k-token vocabulary, and denote split word pieces with $\#\#$. In terms of pre-trained word embeddings, we use *glove.6B*⁶ which is trained with 6B tokens and a 400k vocabulary from the *Wikipedia 2014 + Gigaword 5* corpora. We used public domain implementations^{7,8} throughout our experiments which were run on four (4) Tesla M60 GPUs. It takes about 10 minutes to fine-tune the BERT-SBD model, and about 40 minutes to train the BiLSTM-CRF model.

4.4 Evaluation Metrics

Because the beginning (BS) and ending (ES) tokens⁹ of sentences are evaluated separately, the official FinSBD-2019 evaluation metrics include 1) *F1* scores for predicting BS and ES tokens separately as well as 2) the mean of two separate *F1* scores. We refer to the latter as a soft (lenient) evaluation metric. During training and validation, we used standard evaluation metrics – *Precision* (P), *Recall* (R), and *F1* score – to evaluate BS and ES.

⁶<http://nlp.stanford.edu/data/glove.6B.zip>

⁷https://github.com/guillaumegenthiel/sequence_tagging

⁸<https://github.com/kamalkraj/BERT-NER>

⁹Official FinSBD-2019 annotations use BS for sentence beginning and ES for sentence ending.

To extract well-formed sentences, both beginning and ending tokens need to be predicted accurately. We accordingly propose an additional harsh evaluation metric – *PairSE* – based on the use of *P*, *R*, and *F1* in information retrieval. *PairSE* considers the predicted boundary to be correct only when both BS and ES are correct, calculated as:

$$P = \frac{\{\text{Correct pairs of S and E}\}}{\{\text{All predicted pairs of S and E}\}}$$

$$R = \frac{\{\text{Correct pairs of S and E}\}}{\{\text{All ground truth pairs of S and E}\}}$$

$$F1 = \frac{2 \times P \times R}{P + R}$$

Consider the example sentence “*The company made £10k during 2015 .*” in Section 2: When “*The*” is predicted as S and “*.*” as E, the pair is counted as a correct prediction for *PairSE* ; if either prediction is incorrect or missing, the pair is counted as an incorrect prediction.

4.5 Results and Analysis

Table 3, which includes both the official and our harsh *PairSE* evaluation metric, shows our performance on the validation set with different parameter settings.

System	Class	Official			<i>PairSE</i>		
		<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
BiLSTM1	S	84.3	91.6	87.8			
	E	88.9	96.7	92.6	78.7	85.6	82.0
	Avg	86.6	94.1	90.2			
BiLSTM2	S	82.8	91.8	87.0			
	E	88.6	98.3	93.2	78.4	87.1	82.5
	Avg	85.7	95.1	90.1			
BERT1	S	89.5	94.4	91.8			
	E	91.6	97.4	94.4	86.1	91.6	88.8
	Avg	90.5	95.9	93.1			
BERT2	S	89.4	94.9	92.1			
	E	92.5	98.3	95.3	86.9	92.4	89.6
	Avg	91.0	96.6	93.7			

Table 3: Experimental results on the validation set.

In Table 3, we refer to our systems with different settings regarding the maximum input length. BiLSTM1 stands for the BiLSTM-CRF system in which the input length is limited to 100. The input sequence length is set to 60 for BiLSTM2, which is also denoted as AIG2 in our submissions. BERT1 is the BERT-SBD system where the input length is constrained to 128 while BERT2 is set to 256 in terms of the input length (denoted as AIG1 in our submissions). In addition, Avg represents the MEAN of scores for the corresponding S and E.

On the basis of these results, we conclude that

- Two BERT systems significantly outperformed two BiLSTM-CRF systems across all evaluation metrics.
- BERT2 dominated in terms of *F1* scores.
- Although BiLSTM1 and BiLSTM2 achieved similar scores regarding the official *F1* measure, BiLSTM2 performed better than BiLSTM1 in terms of our harsh *PairSE* metric. We therefore chose BiLSTM2 as one of our submission systems.

- BiLSTM1 obtained higher *P* and lower *R* levels compared to BiLSTM2, which demonstrates that LSTMs can learn long-distance dependencies for predicting sentence boundaries correctly given a long (enough) input sequence.
- For all systems, *Recall* was higher than *Precision* - we suspect our systems are prone to committing to a sentence boundary in ambiguous cases.
- All systems obtained higher scores on E than S which indicates that the ending of a sentence is easier to predict than the beginning, presumably due to the greater frequency of sentence-final punctuation characters (such as the period) and greater diversity of sentence-initial characters.

Table 4 shows our results on the test set for both the official evaluation metrics and our harsh *PairSE* one.

System	Official			<i>PairSE</i>		
	BS	ES	<i>MEAN</i>	<i>P</i>	<i>R</i>	<i>F1</i>
AIG2	0.83	0.88	0.855	71.9	86.6	78.6
AIG1	0.88	0.89	0.885	78.5	90.2	84.0

Table 4: Experimental results on the test set.

AIG1 is our BERT-SBD system with input length 256, and AIG2 is our BiLSTM-CRF system with input length 60. Our AIG1 submission, which is significantly better than AIG2, ranks first amongst all submitted systems in terms of *MEAN F1* score.

5 Conclusions and Future Work

We have described the entry by Investments AI at AIG (American International Group, Inc.) to the FinSBD-2019 shared task (English track). We experimented with two neural systems - BiLSTM-CRF and BERT. We approached sentence boundary detection as a sequence labelling problem, and applied a BIO (IOB) tagging scheme to sentence-initial, -final, and -internal tokens to enrich FinSBD-2019 training data and to train our systems. We fine-tuned our systems with different hyper-parameter settings, and chose BERT-SBD with input length 256 and BiLSTM-CRF with input length 60 for our final submission to the shared task.

Our experimental results on the validation set to date show that our BERT-SBD system performs significantly better than the BiLSTM-CRF variant regarding both the official and our harsher *PairSE* metric. AIG Investments AI BERT-SBD system achieved the highest *MEAN F1* score in the shared task. Our approach and results motivate further research into the use of pre-trained language models for sentence boundary detection. In the future, we will explore more detailed error analyses, evaluate the performance of our SBD systems on even noisier financial documents.

Acknowledgments

The authors would like to thank the reviewers for their insightful comments, Guruprasad Sethurathinam for setting up Amazon Web Services (AWS) for our work, and AIG internal reviewers for providing useful feedback.

References

- [Ait Azzi *et al.*, 2019] Abderrahim Ait Azzi, Houa Bouamor, and Sira Ferradans. The finsbd-2019 shared task: Sentence boundary detection in pdf noisy text in the financial domain. In *The First Workshop on Financial Technology and Natural Language Processing (FinNLP 2019)*, Macao, China, 2019.
- [Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 EMNLP*, pages 1724–1734, October 2014.
- [Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [Evang *et al.*, 2013] Kilian Evang, Valerio Basile, Grzegorz Chrupała, and Johan Bos. Elephant: Sequence labeling for word and sentence segmentation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1426, October 2013.
- [Gillick, 2009] Dan Gillick. Sentence boundary detection and the problem with the U.S. In *Proceedings of NAACL*, pages 241–244, June 2009.
- [Gimpel *et al.*, 2011] Kevin Gimpel, Nathan Schneider, Brendan O’Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th ACL*, pages 42–47, June 2011.
- [Griffis *et al.*, 2016] Denis Griffis, Chaitanya Shivade, Eric Fosler-Lussier, and Albert M. Lai. A quantitative and qualitative evaluation of sentence boundary detection for the clinical domain. In *In Proceedings of AMIA Joint Summits on Translational Science*, pages 88–97, 2016.
- [Halliday, 2004] M.A.K. Halliday. *An Introduction to Functional Grammar*. Arnold, London, United Kingdom, 2004.
- [Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
- [Huddleston and Pullum, 2002] Rodney Huddleston and Geoffrey K. Pullum. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge, United Kingdom, 2002.
- [Kiss and Strunk, 2006] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *American Journal of Computational Linguistics*, 32(4):485–525, 2006.
- [Loper and Bird, 2002] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, pages 63–70, 2002.
- [Ma and Hovy, 2016] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, August 2016.
- [Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL*, pages 55–60, 2014.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, 2013.
- [Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543, 2014.
- [Read *et al.*, 2012] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, December 2012.
- [Reynar and Ratnaparkhi, 1997] Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19, March 1997.
- [Savelka *et al.*, 2017] Jaromir Savelka, Matthias Grabmair, and Kevin D. Ashley. Sentence boundary detection in adjudicatory decisions in the united states. *Traitement Automatique des Langues*, 58(2):21–45, 2017.
- [Schuster and Paliwal, 1997] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, November 1997.
- [Stamatatos and Fakotakis, 1999] Efstathios Stamatatos and N Fakotakis. Automatic extraction of rules for sentence boundary disambiguation. In *Proceedings of the Workshop on Machine Learning in Human Language Technology, Advance Course in Artificial Intelligence (ACAI99)*, pages 88–92, 1999.
- [Tjong Kim Sang and De Meulder, 2003] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, 2017.
- [Wu *et al.*, 2016] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, and et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

ai ai at FinSBD task: Sentence Boundary Detection in Noisy Texts From Financial Documents Using Deep Attention Model

Ke Tian¹, Zi Jun Peng²

¹Office of CTO, OPT, Inc., Japan

²School of Computer Science and Technology, Harbin Institute of Technology(Weihai), P.R.China

¹tianke0711@gmail.com, ²986320586@qq.com

Abstract

This paper describes how we tackle the FinSBD-2019 shared task in IJCAI-2019. The deep attention model based on word embedding is proposed to detect the sentence boundary in noisy English and French texts extracted from the financial documents. The experiment is shown that the model has good performance for predicting beginning and end index of sentence in the two tasks which the result achieved the score of F1 for BS, ES are 0.88,0.91 respectively in English task, and score of F1 for BS, ES are 0.91, 0.92 respectively in French task.

1 Introduction

The first step of many language tasks, such as POS tagging, discourse parsing, machine translation, etc., is the sentence boundary detection (SBD), which detects the end of the sentence [Nagmani Wanjaria, 2016]. This makes the task of detecting the beginning and ending very important, which helps in processing the written language text. However, detecting the end of the sentence is a complicated task due to the ambiguity of punctuation and words in the sentence [Gregory Grefenstette, 1994]. For example, punctuation marks like "." and "!" don't always represent the end part of sentence text and have several functions. The "." can be part of a number like 2.34 or an abbreviation of a phrase, and "!" can represent a word of surprise or shock. A number of research pieces in sentence boundaries mainly used the machine learning methods, such as the hidden Markov model [Mikheev, 2002], Maximum entropy [Jeffrey C.Reynar, 1997], conditional random fields [Katrin Tomanek, 1997], and neural networks [Tibor Kiss, 2006]. Recently, deep learning models have been applied to solve this issue and achieve good performance [Carlos-Emiliano Gonzalez-Gallardo, 2018] [Carlos Emiliano Gonzalez Gallardo, 2018]. Until now, research about SBD has been confined to formal texts, such as news and European parliament proceedings, which have high accuracy using rule-based machine learning and deep learning methods due to the perfectly clean text data. There is no research about the SBD in noisy text that was extracted from the files in machine-readable formats. The FinNLP workshop in IJCAI-2019 is the first proposal of FinSBD-2019 shared tasks that detect sentence boundary in noisy text of finance documents [A Ait Azzi, 2019].

The purpose of FinSBD-2019 shared tasks is to detect the beginning and ending parts of sentences in the noisy text

extracted from financial pdf documents in two languages: English and French. As shown in the English text T1, the provided dataset is a json file containing "text" that has been word tokenized using NLTK, and begin_sentence and end_sentence correspond to all indexes of tokens marking the beginning and the ending of well-formed sentences in the text.

```
T1: [{'text': "Invesco Funds , SICAV\n Vertigo Building  
– Polaris\n Prospectus\n 2 - 4 rue Eugène Ruppert\n L - 2453  
Luxembourg\n 20 August 2013 An open - ended umbrella  
investment fund established under the laws of Luxembourg and  
harmonised under .....",  
'begin_sentence': [22, 50, 79, 120, 128, 1240, 1290, 1315, 1  
344, 1354,.....],  
'end_sentence': [49, 78, 119, 125, 156, 1289, 1314, 1343, 1  
353, 1397,....]}]
```

The goal of the task is to detect the beginning and ending index of English and French sentence text that has been tokenized. We observed that the critical words in the sentence clearly indicated the beginning and ending part of a sentence. For example, in the English text, most of the time, '!', ';' and et al. are at the end of a sentence. The attention mechanism is useful for detecting the weights of words in NLP tasks [Ke Tian 2019]. Therefore, the word2vec-based deep attention model is proposed to detect the beginning and ending index in English and French sentence texts.

Section 2 explains the details of our methods. Section 3 shows experimental configurations and discusses the results. Then, we conclude this paper in Section 4.

2 Deep Attention Model

The structure of the proposed method for detecting the beginning and ending index of sentence texts in English and French is shown in Figure 1. The creating training data and word embedding of English and French texts are first described in Section 2.1. The attention of the long short-term memory (LSTM) [Sepp Hochreiter, 1997] model is described in Section 2.2, and the ensemble result is presented in Section 2.3.

2.1 Recreating Train Data and Word Embedding

In the provided train, dev, and test data in English and French, the words have been tokenized. We observed that the end part of a sentence does not just use punctuation like '.' and ';', and includes some words like 'as' and 'and', which caused the ending part be complex. Like the ending part, the beginning

part of sentence also is not just words which beginning with upper letter like ‘The’, ‘Given’, ‘This’, also include symbol character like ‘(’, ‘-’. Therefore, using only the rule to detect the beginning and end of a sentence may be not useful. We found that the unusual beginnings and endings are identifiable by context. Moreover, we found that the provided training data was not easy to use for the deep learning model. We recreated the new training data that can be applied to deep learning model based on provided train, and dev data. The procedure for recreating the new training data is shown in Fig 2.

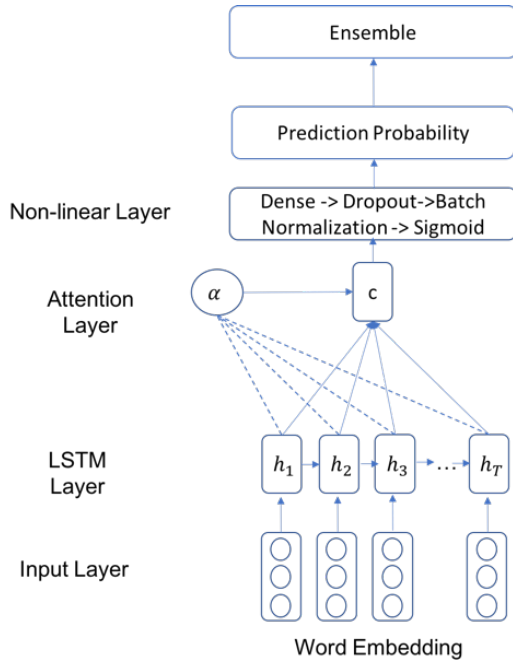


Figure 1: Deep attention model

We take the T1 sentence as an example to describe how to recreate the new train data. As the provided T1 JSON data, each tokenized word is labeled. For example, ‘Invesco’, ‘Funds’, ‘An’, ‘amended’, and ‘2016’ are labeled ‘O’, ‘O’, ‘BS’, ‘ES’, and ‘O’ respectively. As each tokenized word, the previous n tokenized words following n tokenized words of each tokenized word are taken to concatenate into a new sentence. For example, take the 5 previous words as an example. As the first word is "Invesco" in the T1, there are no previous 5 words, so we added 5 "pre" words at the beginning of the sentence. Therefore, the new sentence for "Invesco" is the T2 sentence. With the beginning word "An", the new sentence is T3. As the end word of T1, there are no next 5 words, so we added 5 "EOS" words at the end of the sentence. Therefore, the new sentence for "2016" is T4. The labels of T2, T3, and T4 are "O", "BS", and "O" respectively, which are the same as the labels of the tokenized words "Invesco", "An", and "2016" respectively. The train, dev, and test data in English and French both use the same method to recreate data. There are three labels (O, BS, ES) for the train, dev, and test data. Therefore, the goal of the task is changed to classify the labels of new data.

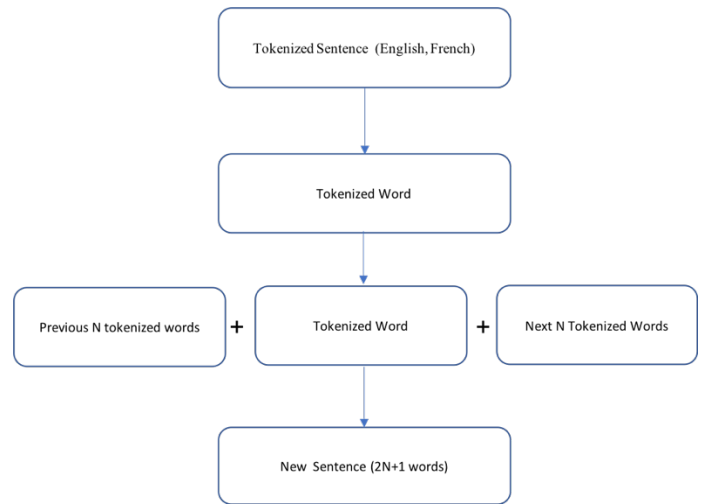


Figure 2: Procedure of recreating data

- T2: Pre Pre Pre Pre Pre Invesco Funds , SICAV \n Vertigo Building.
- T3: 2453 Luxembourg\n 20 August 2013 An open - ended umbrella investment.
- T4: II – Version of APRIL 2016 EOS EOS EOS EOS EOS

Word embedding is the foundation of deep learning for natural language processing. We use the new train, dev, test text data to train the word embedding. At the beginning, the words are not converted into lowercase. In the recreated English text data, there are 1010868 recreated sentences with 14285 unique token words from the training, dev, and test data. In the French text, there are recreated 1053437 sentences with 15784 unique token words from the training, dev, and test data. The CBOW model [Tomas Mikolov, 2013] is taken to train word vectors for the English and French text data, and the word2vec dimension is set to 100.

2.2 Attention-based LSTM Model

Through the task train data, we observe that some keywords could help decide the category of a sentence. For example, ".", ";", and "as" indicate the ending part of sentence. The ES category is indicated by the keywords "The" or "This". Thus, some keywords in the sentence have more importance to predict the label of sentence text. Since the attention mechanism can enable the neural model to focus on the relevant part of your input, such as the words of the input text, the attention mechanism is used to solve the task. In this paper, we mainly use the feed-forward attention mechanism [Colin Raffel, 2015]

The attention mechanism can be formulated with the following mathematical formulation:

$$e_t = a(h_t), \alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^T \exp(e_k)}, c = \sum_{t=1}^T \alpha_t h_t$$

In the above mathematical formulation, a is a learnable function and only depend on h_t . The fixed-length embedding c of the input sequence computes with an adaptive weighted average of the state sequence h to produce the attention value.

In the structure of the proposed model, as the LSTM layer, the embedding dimension and max word length of word embedding are set to be 100 and $2n+1$ (n is the number words surrounding the tokenized word), respectively, as the embedding dimension. The embedding layer of the word embedding matrix as an input layer of LSTM and the size of the output dimension is 300. We used the feed-forward attention mechanism as the attention layer. As the non-linear layer, the activation function is to dense the output of the attention layer to be 256 dimensions, and by using the dropout rate of 0.25, the output result after the dropout rate will be batch normalization. Finally, the sigmoid activation function that will dense the dimension of batch normalization input will be the length of the label as the final output layer.

2.3 Ensemble Result

In the model training stage, the 10-fold cross validation is used to train the deep attention model for predicting the test data. We sum 10 folds of predict probability and get the mean value of 10 folds for the final predict probability result. In the task, two results for each language are submitted: one result is based on the word embedding of the deep attention model, and the other result is based on word embedding of the convolutional neural networks (CNN) [Kim, 2014] models.

3 Experiments

3.1 Experiment Design and Implementation

In the experiment, rule-based, CNN and the proposed deep attention model have been implemented in the task. Moreover, in the data processing stage, keep the upper letter of words to train the word embedding in the English and French text. In addition, we test the different numbers of words surrounding each tokenized word. The numbers 5, 8, 10 are taken to be tested in the experiment.

As the simple rule-based method in the experiment. In the English task, we just determined the end index by the '!', '\n', '!', ':', ';' token words, and the beginning index is detected by the word that the beginning character of token word is upper letter or the token word is '(' . As the French task, the beginning index is determined by the word that the beginning character of token word is upper letter, and the end index is detected by the '!', '\n', '!', ':', ';' token words.

The 10-fold cross-validation to predict the test data is used in the model. The deep model in our research was implemented with Keras [Keras, 2019]

Based on the evaluation requirements of the FinSBD Task, the F-scores are taken to evaluate the performance of the proposed model in the paper.

3.2 Result and Discussion

The results of rule-based, CNN, LSTM and the deep attention model are shown as Table 2. From Table 2, as the English task, the worst performance of model is rule-based which the F1 score of ES, BS is 0.17, and 0.72. Moreover, the F1 score of ES, BS of CNN model is 0.82 and 0.89. The F1 score of BS, and ES of the deep attention model is 0.83 and 0.91, respectively, which is better than the CNN and LSTM model. The F1 score of the ES and BS of the deep attention model is also better than the CNN model and Rule-based in the French task. The result showed that deep attention model is effective to predict the beginning and end index of task sentence in English and French.

Model	English			French		
	BS	ES	Mean	BS	ES	Mean
Rule-based	0.17	0.72	0.445	0.34	0.51	0.425
CNN	0.82	0.89	0.855	0.89	0.90	0.895
Deep Attention	0.83	0.91	0.875	0.91	0.92	0.915

Table 1. Experiment Results: F-score of English, French tasks using Rule-based, CNN and Attention-LSTM, and the surrounding number of words is 5

N words	English			French		
	BS	ES	Mean	BS	ES	Mean
5	0.83	0.91	0.875	0.91	0.92	0.915
8	0.86	0.90	0.88	0.91	0.92	0.915
10	0.88	0.91	0.895	0.91	0.92	0.915

Table 2. Experiment Results: F-score of English, French tasks, and the surrounding number of words is 5, 8,10 respectively for deep attention model

In order to test how the number of surrounding words affect performance, the 5, 8, and 10 surrounding words for the deep attention model were implemented in the experiment, and the result is shown in Table 2. Based on the result, the best performance of SBD prediction is the 10 surrounding numbers for tokenized words in which the F1 score of ES and BS in the English task are 0.88 and 0.91, respectively. However, the 5, 8, and 10 surrounding words for the deep attention model in the French task, the score of F1 is the same.

Therefore, we may infer that the more words surround the tokenized word, the better the prediction is in English task. As the French task, the number of words surrounding the tokenized word may not influence the performance of prediction in this task.

Team	ES	BS	Mean
AIG1	0.88	0.89	0.885
seernet1	0.85	0.9	0.875
aiai1	0.83	0.91	0.87
isi1	0.83	0.89	0.86
NUIG1	0.81	0.9	0.855
isi2	0.82	0.89	0.855
AIG2	0.83	0.88	0.855
AI_Blues2	0.82	0.87	0.845
AI_Blues1	0.82	0.87	0.845
mhirano1	0.78	0.89	0.835
aiai2	0.79	0.88	0.835
NUIG2	0.81	0.85	0.83
HITS-SBD2	0.8	0.86	0.83
HITS-SBD1	0.8	0.86	0.83
PolyU_CBS-CFA_NN1	0.77	0.86	0.815
PolyU_CBS-CFA_RFC1	0.7	0.86	0.78
PolyU_CBS-CFA_RFC2	0.68	0.86	0.77
mhirano2	0.58	0.67	0.625

Table 3 Leaderboard FinSBD in English task

Based on the final report about FinSBD-2019 shared tasks as shown in the Table 3 and Table 4. As the English task, the ranking of our team is 3, and aiai1 is the result of deep attention model. As the French task, aiai1 is ranked 2, and some indicators such as ES rank number 1. Due to my busy schedule, we only submitted 5 words around the tokenized word before the deadline, and we found that there is code in error in CNN model in last submission which caused the abnormal score (aiai2) in these tasks. Currently, the updated result is shown in the Table 2. Based on Table 2, if we take the result of 10 surrounding words for submission, our team would rank number 1 in the English task. The result showed that the proposed model could effectively predict the beginning and

ending indexes of words in the noisy text of finance documents in these two tasks.

Team	ES	BS	Mean
seernet	0.91	0.93	0.92
aiai1	0.91	0.92	0.915
NUIG1	0.9	0.92	0.91
NUIG2	0.9	0.92	0.91
isi1	0.9	0.91	0.905
isi2	0.89	0.91	0.9
AI_Blues1	0.85	0.88	0.865
AI_Blues2	0.84	0.88	0.86
PolyU_CBS-CFA_RFC1	0.84	0.88	0.86
mhirano1	0.82	0.89	0.855
PolyU2	0.83	0.87	0.85
PolyU_CBS-CFA_NN1	0.83	0.87	0.85
PolyU_CBS-CFA_RFC2	0.81	0.88	0.845
mhirano2	0.67	0.68	0.675
aiai2	0.01	0.02	0.015

Table 4 Leaderboard FinSBD in French task

4 Conclusion

This paper mainly discusses how we tackle the FinSBD-2019 shared task. There are two tasks which predict beginning and ending index of words in the sentence text of finance document in English and French. In order to tackle these tasks, we firstly recreate the train, dev, and test data so that can be applied to deep learning model. Then, the deep word embedding-based attention model is proposed to classify the labels of recreated data. The experimented result showed that the proposed model could effectively solve the goal of task and achieve very good performance in these tasks.

References

[Abderrahim, 2019] Ait Azzi Abderrahim, Bouamor Houda, and Ferradans Sira. The FinSBD-2019 Shared Task: Sentence boundary detection in PDF Noisy text in the Financial Domain, The First *Workshop on Financial Technology and Natural Language Processing of IJCAI 2019*:16–19, Macao, China, August 2019.

- [Carlos-Emiliano Gonza lez Gallardo1, 2018] Carlos-Emiliano Gonza lez Gallardo1 and Juan-Manuel Torres-Moreno. Sentence boundary detection for french with sub-word level information vectors and convolutional neural networks. <https://arxiv.org/abs/1802.04559>, 2018.
- [Carlos-Emiliano Gonza lez-Gallardo, 2018] Carlos-Emiliano Gonza lez-Gallardo, Elvys Linhares Pontes, Fatiha Sadat, Juan-Manuel Torres-Moreno. Automated sentence boundary detection in modern standard arabic transcripts using deep neural networks. *Procedia Computer Science*, (142):339–346, 2018.
- [Colin Raffel, 2015] Colin Raffel and Daniel P. W. Ellis. Feed-forward networks with attention can solve some long term memory problems. <https://arxiv.org/abs/1512.08756>, 2015.
- [Gregory Grefenstette, 1994] Gregory Grefenstette and Pasi Tapanainen. What is a word, what is a sentence? problems of tokenization. In *In Proc.3rd International Conference on Computational Lexicography (COM- PLEX'94)*, pages 79–87, 1994.
- [Jeffrey C.Reynar, 1997] Jeffrey C.Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, pages 16–19, Washington, USA, April 1997.
- [Katrin Tomanek, 1997] Katrin Tomanek, Joachim Wermter and Udo Hahn. Sentence and token splitting based on conditional random fields. In *Proceedings of the 12th World Congress on Health (Medical) Informatics*, pages 16–19, Washington, USA, April 1997.
- [Ke Tian, 2019] Ke Tian and Zi Jun Peng. aiai at FinNum task: Financial numeral tweets fine-grained classification using deep word and character embedding-based attention model. *The 14th NTCIR Conference*, Tokyo, Japan, June 2019.
- [Keras, 2019] Keras. The python deep learning library. <https://keras.io>. Accessed: May 2019
- [Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. <https://arxiv.org/abs/1408.5882>, 2014.
- [Mikheev, 2002] Andrei Mikheev. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):289–318, September 2002.
- [Nagmani Wanjaria, 2016] Nagmani Wanjari, G. M. Dhopavkar Prof and Nutan B. Zungre. Sentence boundary detection for marathi language. *Procedia Computer Science*, 78:550–555, 2016.
- [Sepp Hochreiter, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735-1780, 1997.
- [Tibor Kiss, 2006] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, (32):485–525, April–June 2006.
- [Tomas Mikolov, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. <https://arxiv.org/abs/1310.4546>, 2013.

Pluto: A Deep Learning based Watchdog for Anti Money Laundering

Hao-Yuan Chen¹, Shang-Xuan Zou¹, Cheng-Lung Sung¹

¹Data Research & Development Center, CTBC Bank, Taipei, Taiwan
{haoyuan.chen, mi.zou, alan.sung}@ctbcbank.com

Abstract

Banks are faced with Anti-Money Laundering (AML) obligations so they have to identify customers by conducting negative news, aka "adverse media", screening which consists of searching information in the public domain for news items, publications, and government advisories and bulletins for information related an individual or entity's involvement in financial crime matters. Although it is an essential way to determine who in the group poses a higher risk for potential financial crime concerns by catching sophisticated activities from negative news across globe, it also requires heavy human capital and processing time on screening daily produced negative news. Therefore, poor efficiency becomes the most unacceptable obstacle based on this approach. To mitigate this issue, Pluto¹ offers a distributed and scalable batch system embedded deep learning-based Natural Language Processing (NLP) techniques for AML practitioners to improve daily task efficiency. It performs *text preprocessing*, *paragraph embeddings*, and *clustering algorithm* on a set of negative news and provide clustering result with keywords and similarities for AML practitioners. The overall feedback from AML practitioners are very positive on such an impressive enhancement in which Pluto reduces 67% efforts of negative news screening.

1 Introduction

Compliance with Anti-Money Laundering (AML) and Know Your Customer (KYC) duties require identity checks against potential money launderers, terrorists, or people considered high risk in the financial field. Although the concept is reasonable and feasible, there are tremendous challenges on insufficient approaches capable of offering an effective and efficient solution. It triggers lots of techniques been proposed in real world industry [Han *et al.*, 2018].

We leverages GlobalVision's Patriot Officer² to discover whether customers are high-risk individuals or not via check-

ing on a variety of watchlists (e.g., Dow Jones Watchlist) across the globe. One of the required steps is to screen plenty of related documents associated with several watchlists which are related to a single customer. It is an effective method to identify hidden criminal activities but not an efficient one. The volume of daily-crawled content are sometimes greater than the ones consumed by practitioners. It makes practitioners engage in investigation with massive human capital cost and processing time. Therefore, improving operational efficiency is the top priority for such an approach.

We propose an idea of on-demand document clustering through client-server model batch processing system in order to alleviate suffering in practitioners' daily task. It aims to group a set of documents in a way we proposed so that the documents in the same group are similar to each other than to those in other groups. It provides a web-based user interface (UI) not only revealing clustering result in tree-view structure accompanied by related keywords but showing similarity between the latest one and the others within each group as well. The purpose is to support practitioners conducting an effective decision-making in such an efficient way.

2 Design and Implementation

In this section, we introduce the details on Pluto architecture and method based on NLP and fundamental algorithms. It only supports the articles published in Traditional Chinese (zh-TW) at this stage.

2.1 Architecture

The client-server model batch processing system follows a micro-service oriented distributed architecture [Fehling *et al.*, 2014] presented in Figure 1.

The *API* service provides REST endpoints for the interaction between the *UI* and the *Patriot Officer*. It triggers processing pipelines by sending clustering job into *AMQP*³ when receiving requests from *Patriot Officer* and provides the clustering result information for *UI* from *DB*. The *Workers* service fetches jobs from *AMQP* to proceed pipeline for document clustering based on proposed solutions. The *UI* allows

¹A cartoon dog created in 1930 at Walt Disney Productions.

²<http://www.gv-systems.com/products-solutions/patriot-officer/>

³AMQP is an open standard application layer protocol used for queuing and routing messages between the services in a secure and reliable way

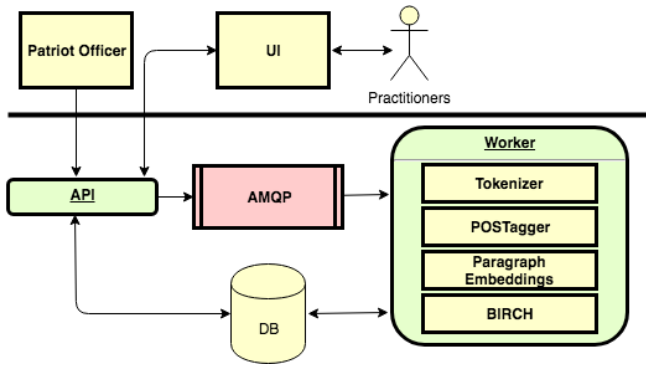


Figure 1: Pluto Architecture

practitioners to retrieve clustering information related to the specific watchlist.

The batch processing system is scalable. The quantity of *Workers* can be scale up and down according to quantity of jobs residing in AMQP, which makes the system achieve throughput guarantee.

2.2 Method

The pipeline comprises three sequential components: text preprocessing, paragraph embeddings construction, and clustering. Input is a set of documents associated to a single watchlist, and the output is a set of clusters containing all documents and the similarities within a single cluster.

Initially, we pick up tokens that match keywords offered by AML practitioners or be tagged as human names and proprietary nouns such as political groups, administrative unit, facilities, locations, organizations, and etc. Sinica CKIP tokenizer and POS tagger [Ma and Chen, 2003] are the major NLP techniques on this topic. The extracted tokens are regarded as representative for each document.

Next, we build the vector representation by *Distributed Bag of Words version of Paragraph Vector (PV-DBOW)* model [Le and Mikolov, 2014] which not only being conceptually simple but also requires to store less data. Moreover, it ignores the context words, which aligns the empirical insights from AML practitioner. The adopted deep learning-based techniques revealed from [Mikolov *et al.*, 2013a] and [Mikolov *et al.*, 2013b] is implemented by Gensim⁴ for this stage.

Eventually, we infer vectors for documents based on paragraph embeddings and apply *Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH)* [Zhang *et al.*, 1996] to proceed clustering. The reason why we adopt it is that BIRCH, one of the developments in hierarchical clustering, does not require us to pre-specify the number of clusters and is a memory-efficient and time-efficient approach. [Xu and Tian, 2015]

3 Evaluation

To show the effect of Pluto, we cooperate with professionals of AML to quantitate the performance via user testing. We

⁴<https://radimrehurek.com/gensim/index.html>

divide users into 2 groups: (1) reading news without clustered news and (2) reading news with Pluto.

For Pluto users, we implement a web interface to show the performance difference. As shown in Figure 2, the original system provides only an URL list, while Pluto visualizes negative news into clusters with keywords and similarities. With the interface, users can comprehend the news efficiently, and determine the suspect customer fast.

To evaluate the performance, both groups read the same news set contains 3000 news. As a result, Pluto reduces 67% time in average to judge if the customer suspected of monev laundering.

編號	來源
1	Taipei Times_12-Nov-2008_on Factiva.com (https://global.factiva.com/redirect/default.aspx?P=sa&AN=TAIP000020081117e4bc0000&cat=a&op=ASE)
2	The Wall Street Journal Online_05-Jan-2015_on Factiva.com (https://global.factiva.com/redirect/default.aspx?P=sa&AN=WSJ0000020150105ab15002e5&cat=a&op=ASE)
3	Ming Pao Instant News (Chinese Language - Simplified)_11-Sep-2009_on Factiva.com (https://global.factiva.com/redirect/default.aspx?P=sa&AN=MNPACN0020090911e59b018&cat=a&op=ASE)
4	http://irs.judicial.gov.tw/GNNWS/NNWS002.aspx?id=57279&flag=1&key=8&muchinfo=&countid=

(a) Original User Interface

(b) Pluto User Interface

Figure 2: Pluto offers grouped negative news annotated with keywords and similarities. In Figure 2b, the left side is tree-view structure for navigation purpose, and the right side is the pane for annotated negative news.

4 Conclusion and Future Work

In this work, we propose a distributed architecture batch system based on NLP techniques to organize daily negative news and offer an UI for information visualization. At present, the entire system is at piloting stage in our private cloud and facilitates efficient work flow among AML investigations pipeline.

In the future, we will (1) adopt our system to multilingual use cases, especially including Simplified Chinese (zh-CN) and English (en). (2) utilize NLP techniques in further investigation in which we may embrace named entity recognition (NER) and relation extraction (RE) to build the relation network identifying target suspicious entities, events, and time, and location.

References

- [Fehling *et al.*, 2014] Christoph Fehling, Frank Leymann, Ralph Retter, Walter Schupeck, and Peter Arbitter. *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*. Springer, 2014.
- [Han *et al.*, 2018] Jingguang Han, Utsab Barman, Jeremiah Hayes, Jinhua Du, Edward Burgin, and Dadong Wan. Nextgen aml: Distributed deep learning based language technologies to augment anti money laundering investigation. In *Proceedings of ACL 2018, System Demonstrations*, pages 37–42, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14*, pages II–1188–II–1196. JMLR.org, 2014.
- [Ma and Chen, 2003] Wei-Yun Ma and Keh-Jiann Chen. Introduction to ckip chinese word segmentation system for the first international chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing - Volume 17, SIGHAN '03*, pages 168–171, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
- [Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13*, pages 3111–3119, USA, 2013. Curran Associates Inc.
- [Xu and Tian, 2015] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, Jun 2015.
- [Zhang *et al.*, 1996] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD '96*, pages 103–114, New York, NY, USA, 1996. ACM.

From Creditworthiness to Trustworthiness with alternative NLP/NLU approaches

Charles Crouspeyre – Eléonore Alesi – Karine Lespinasse

Dnalytiks - Live with AI - MITSIC University Paris 8

charles@dnalytiks.com - eleonore.alesi@livewithai.org - karine.lespinasse-sabourault@univ-paris8.fr

Abstract

The word “Credit” comes from the Latin ‘credere’ which means “to give trust”. The way this trust is being given today by financial institutions is mainly based on statistical methods, using financial information - such as the revenues of an individual, the money he/she spends monthly, etc. - to quantify the chances of a successful loan repayment. By definition, this method of quantifying risk limits the access to financial services to individuals who have banking information history. In emerging markets, as described here, a huge part of the population doesn't meet these criteria and is accordingly left outside of the equation. This paper discusses alternative approaches to allow these unbanked people to access financial services in reviewing current and innovative Natural Language Processing and Natural Language Understanding methods. The latter support excellent risk quantification results without affecting the privacy of the borrower and result in moving from biased creditworthiness to broader trustworthiness.

1 Introduction

About 90% of today's global data were created over the last two years alone¹. The most recent estimations indicate that 80% of the total existing data is unstructured². This vast expanse of data contains valuable information that can be used to augment a wide range of processes in the financial sector including fraud detection, market prediction, customer relationship management but also credit scoring.

Contemporarily, the way credit risk is being assessed by financial institutions relies heavily on financial information, or creditworthiness, such as the FICO scores. In the USA,

for example, over 90% of top lenders use FICO Scores as their credit scores³. The makers of the FICO scores have not publicized how the scores are calculated⁴. But they disseminated the weights of the criteria that they look at as: 35% payment history, 30% amount owed, 15% length of history, 10% new credit, 10% types of credit used. Consequently, as they can only provide little of this type of information, “two billion of adults - more than half of the world's working adults - are still excluded from formal financial services” according to the UNCDF Annual report⁵.

Such scoring models are even less applicable to emerging markets, since the majority of the population doesn't possess any type of financial information. For example, in the Philippines, 77% of the population is considered unbanked -i.e. doesn't have a bank account, according to the BSP, the Central Bank of the Philippines⁶.

Aligned with the United Nations' Financial Inclusion pillar, one of the key pillars required to reach its 17 Sustainable Development Goals, this article aims at exploring NLU-based viable alternative forms of risk analysis to enable greater access to credit.

2 Creditworthiness evaluated via Machine Learning approaches

Recent advances in Machine Learning, and more broadly in Artificial Intelligence [Bach et al., 2019], have enabled the emergence, in the last few years, of alternative methods for qualifying and quantifying the creditworthiness of an individual, using non-financial information innovative

¹ - <https://www.ibm.com/products/software>

² - <https://www.techrepublic.com/article/unstructured-data-the-smart-persons-guide>

³ - <https://www.fico.com/en/products/fico-score>

⁴ - <http://money.com/money/3770477/new-fico-score-factors>

⁵ - <https://www.uncdf.org/financial-inclusion>

⁶ - <http://nine.cnnphilippines.com/business/2018/07/11/2017-financial-inclusion-survey-bsp.html>

approaches such as phone log analysis [Shema, 2019] or [Agarwal et al., 2018] or social network analysis⁷. A few startups have emerged from those efforts, and have displayed encouraging results in helping solving the problem of financial inclusion for emerging markets including Lenddo in Singapore, previously mentioned, or Sesame Credit in China.

Nevertheless, these approaches tend to be invasive, with little care over the users' privacy, e.g. requiring the sharing of personal contacts with the system, etc. For this reason, another approach, based on Natural Language Processing (NLP) & Natural Language Understanding (NLU) tools, could enable a new baseline to quantifying risk with regards to predicting borrower repayment behavior and exclusively accessing information directly related to the user. This approach would also allow a change of paradigm, from scoring the Creditworthiness of an individual to scoring his Trustworthiness.

Creating NLP tools that preserve privacy has been addressed in the healthcare industry. The US National Library of Medicine's Lister Hill National Center for Biomedical Communications⁸ uses NLP to "de-identify" clinical information in narrative medical reports, protecting patient privacy while preserving clinical knowledge.

In this specific case, NLP has been used to ensure that all the information related to individual private information can be identified and then "de-identified."

3 Trustworthiness evaluated via NLU approaches

In the last few years, NLP and NLU have gained a lot of traction across industries, thanks to the progress made in Deep Learning and the capability it has brought along to analyze text at a semantic level - with discoveries such as word embeddings [Mikolov et al., 2013], when previous approaches were mostly limited to statistical analysis.

This new capacity enabled a deeper level of analysis, allowing different levels of understanding of a text document, for example combining the content itself with the non-explicit information such as emotions. Such capacity is being used by call centers for example to detect sentiment such as anger and stress through analysis of sentences' structures and the choice of words [Kumar et al., 2012] or [Seyeditabari et al., 2018] in order to adapt answers provided by the representative to a request, in real time.

This NLP-based approach, generalized to credit scoring, allows to bridge the gap and understand hidden patterns that cannot be humanly detected. It relies on many more data points than usual decision-making processes, providing

better estimates and enabling to identify patterns requiring further attention from the decision maker, while getting a more nuanced view of trustworthiness with an improved rating accuracy of loans.

It could, among other criteria, help to:

- Detect & score business knowledge of the borrower by understanding the person's entrepreneurial ability and attitude toward financial planning. This approach has been successfully implemented by platforms such as Capital Float and Microbnk;
- Detect fraud attempt by automatically detecting incoherencies in the information provided by the borrower [Iter et al., 2018];
- Detect attempts to convey alternative messages: as discovered by [Netzer et al., 2018] where data from Peer-to-Peer lending platforms such as Lending Club and Prosper is analyzed, showing that the inclusion of alternative messages - such as religious references, in loan application would lead to poorer repayment rate. A potential explanation of this phenomenon could be an attempt from the borrower to call to the lender's emotions, as a way to falsely convince him of a favorable outcome.

Additionally, as showcased by the company Microbnk in their related patent US20170018030A1, this type of model could even be more powerful if combined with other alternatives forms of data such as psychometric data. For example:

- Counting the number of times, a borrower presses a key Vs. the number of characters in his description;
- Combining NLP with voice tone analysis -among other external parameters.

4 Discussion

In this article, it has been explored how Natural Language Processing/Natural Language Understanding could bring financial solutions to the unbanked populations through innovative approaches. NLP being fundamentally less invasive than requiring direct access to a person's full contact list or social media account, it represents a fairer way to assess risk with a capability to reach a broader audience.

However, monitoring criteria that would indicate, for example, the education level of a person, through grammatical analysis, as a proxy for potentially quantifying the income level of a person would re-inject the initial biases, likewise deteriorating the score of an individual because this person lives in a low-income neighborhood. Particular attention will be required to maximize an ethical methodology in implementing NLP/NLU solutions.

⁷ - <https://patents.justia.com/patent/8694401>

⁸ - <https://lhncbc.nlm.nih.gov/project/de-identification-tools>

References

[Agarwal et al., 2018] RR Agarwal, CC Lin, KT Chen, VK Singh. Predicting financial trouble using call data-On social capital, phone logs, and financial trouble. *PLoS One*, 13(2), February 2018.

[Bach et al., 2019] Mirjana Pejić Bach, Živko Krstić, Sanja Seljan and Lejla Turulja. Text Mining for Big Data Analysis in Financial Sector: A Literature Review. *Sustainability*, 11(5):1277-1304, February 2019.

[Iter et al., 2018] Dan Iter, Jong Yoon, and Dan Jurafsky. Automatic Detection of Incoherent Speech for Diagnosing Schizophrenia. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, pp. 136-146, New Orleans, LA, USA, June 2018. Association for Computational Linguistics

[Kumar et al., 2012] Kumar, M. Naveen, and R. Suresh. Emotion detection using lexical chains. *International journal of computer applications*, 57(4), November 2012

[Mikolov et al., 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111-3119, 2013.

[Netzer et al., 2018] Oded Netzer, Alain Lemaire, and Michal Herzenstein. When words sweat: Identifying signals for loan default in the text of loan applications. *Columbia Business School Research Paper*, pp.16-83. 2018.

[Seyeditabari et al., 2018] Armin Seyeditabari, Tabari Narges, and Zadrozny Wlodek. Emotion Detection in Text: a Review. *arXiv preprint arXiv:1806.00674*, 2018.

[Shema, 2019] Alain Shema. Effective credit scoring using limited mobile phone data. In *Proceedings of the Tenth International Conference on Information and Communication Technologies and Development (ICTD'19)*, 11 p, Ahmedabad, India, January 2019.

On a Chatbot Conducting a Virtual Dialogue in Financial Domain

Boris Galitsky¹ and Dmitry Ilvovsky²

¹Oracle Inc., Redwood Shores, USA

²National Research University Higher School of Economics, Moscow, Russian Federation

Abstract

We demo a chatbot in a personal finance domain that delivers content in the form of virtual dialogues automatically produced from plain texts extracted and selected from documents. Given an initial query, this chatbot finds documents, extracts topics from them, organizes these topics in clusters according to conflicting viewpoints, receives from the user clarification on which cluster is most relevant to her opinion, and provides the content for this cluster. This content is provided in the form of a virtual dialogue where the answers are derived from the found and selected documents and its split results, and questions are automatically generated for these answers.

1 A Virtual Dialogue

Presentation of knowledge in dialogue format is a popular way to communicate information effectively. Usability studies have shown that for information acquirers, dialogues often communicate information more effectively and persuade stronger than monologue most of the times [Cox et al., 1999].

A virtual dialogue is defined as a multi-turn dialogue, possibly with adversarial argumentation, between imaginary agents obtained as a result of content transformation [Piwek et al., 2007].

Dialogue construction from the text is based on the Rhetorical Structure Theory (RST, [Mann and Thompson., 1988]). To obtain a sequence of answers from a text, we automatically split it into elementary discourse units (EDUs) and then form a discourse tree where these units are labels for the leaves of this tree. Satellite EDUs are then selected as answers to the questions that are derived from these EDUs by means of generalization. Nucleus EDUs will follow the satellite EDUs in dialogue answers to maintain its cohesiveness [Barzilay and Lapata, 2008; Galitsky et al., 2015]. Discourse tree is constructed with the RST-parser [Surdeanu et al., 2015].

An example of converting a text into a virtual dialogue is shown in the form of a discourse tree below. Answers are obtained by splitting text into EDUs, and questions are inserted in text before satellite EDUs. Generated questions are shown in angle brackets and bolded.

For the text “*Lower risk investments, such as government bonds, offer lower but more stable returns and less chance of losing the money you invest. But, lower returns may mean that inflation reduces the value of your investments*” we have the following discourse tree with inserted questions:

elaboration (LeftToRight)

contrast (RightToLeft)

same-unit

TEXT: Lower risk investments such as government bonds

elaboration (LeftToRight)

<what kind of return it can give?>

TEXT: offer lower but more stable returns, less chance

elaboration (LeftToRight)

TEXT: of losing the money

TEXT: you invest .

attribution (RightToLeft)

TEXT: But, lower returns may mean

<What can inflation do with my investment?>

TEXT: that inflation reduces the value of investments.

2 A Session with a Virtual Dialogue Chatbot

We present an exploratory session that combines information delivery in the form of traditional textual answers (conventional dialogue) and in a form of a virtual dialogue. The dialogue starts from the user question, ‘*pros and cons of investment in annuity*’ (Fig. 1).

Bot>ask a new question

User> **pros and cons of investment in annuity**

Bot> exploring <https://www.bankrate.com/retirement/5-pros-andcons-of-investing-in-annuities/>

I believe these are the main topics of your query: is that what you meant? Please select

[1] 'establish your own salary from an annuity'

[2] 'an insurance or investment company in exchange'

[3] 'predictable retirement income'

User> **predictable retirement income**

Bot> If you are thinking of investing for retirement years,

there are many options for you to choose from...
*Are you OK with this answer? yes/more/no/specify
 [different topic] // virtual dialogue*

User > **virtual dialogue**

Bot> This is what other people chat about your topic:

Bot>An advantage of an annuity is the reliable income that it produces, according to annuity-pros-cons.com .
 User>**What will my own salary be?**
 Bot>The annuity establishes a known schedule for when you will receive your money. This allows you to establish your own salary from an annuity
 User>**What if I find a better investment?**
 Bot> While an annuity pays you a known amount for a specific amount of time, it cannot be changed. You cannot pull your money out of the annuity.

User> **What category of people has the highest financial anxiety?**

Bot>This has been answered to users with similar questions:

Bot> Having a predictable retirement income can help you feel happier
 User>**Where is the strongest financial anxiety?**
 Bot> Conversely, the researchers discovered that retirees who must withdraw money from investments to pay for retirement expenses had the highest financial anxiety.

Figure 1: Chatbot session with conventional and virtual dialogue.

The chatbot consults the sources and extracts the content from each page (or document) expected to be relevant for the query. The chatbot forms the list of topics extracted from these search results so that the user might select the one of her interest. The user selects his topic of interest and requests a specific answer. Once the answer is read, there are multiple options:

- accept the answer and conclude the session;
- navigate to the next/specific answer from the chatbot list;
- attempt to reformulate the query;
- reduce search to a specified web domain;
- proceed to more search results in the form of a virtual dialogue.

The user selects the last option and the chatbot builds a virtual dialogue. It is a conversation between imaginary people where the conversation topic is retained, matching the original query. Virtual dialogues are shown in frames.

Chatbot demo videos (please, check *10 min video*) and instructions on how to use it are available at our [GitHub](#) in the “**What is new?**” section.

3 Evaluation of Effectiveness

Evaluating the effectiveness of information delivery via virtual dialogues, we compare the conventional chatbot sessions where users were given plain-text answers, and the ones where users were given a content via virtual dialogues.

Table 1. Evaluation of comparative effectiveness of conventional and virtual dialogues

	Conventional dialogues			Virtual dialogues		
	# of iterations till found	# iterations till decision	Coverage of exploration # of entities	# of iterations till found	# iterations till decision	Coverage of exploration # of entities
Conventional only	4.6	6.3	10.8	-	-	-
Virtual only	-	-	-	4.1	6.0	13.7
Conventional followed by virtual	4.0	5.7	7.6	6.1	11.3	15.1
Virtual followed by conventional	5.6	7.1	12.3	3.7	7.0	11.5

We assess dialogues with respect to following usability properties.

The speed of arriving to a decision to commit a transaction such as purchase or product selection. A user is expected to accumulate sufficient information, and this information should be convincing enough for making such decision;

We also measure **how many entities** (in linguistic sense) were explored during a session with the chatbot. We are interested in how thorough and comprehensive the chatbot session is. This assessment is sometimes opposite to the above two measures but nevertheless is important for understanding the overall usability of various conversational modes.

We do not compare precision and recall of search sessions with either dialogue mode since the same information is delivered, but in distinct modes.

In the first and second rows, we assess the stand-alone systems. Virtual dialogues take less iteration on average for information access and about the same number of iterations for decisions as conventional dialogues do.

In the bottom two rows, we observe the usability of the hybrid system. When a conventional dialogue is followed by a virtual one, a lower portion of users is satisfied by the first step in comparison to the inverse architecture, where virtual is followed by conventional.

Acknowledgements

The article was prepared within the framework of the HSE University Basic Research Program and funded by the Russian Academic Excellence Project '5-100'.

References

- [Mann and Thompson, 1988] William Mann and Sandra Thompson. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text* 8(3):243–281. 1988
- [Barzilay and Lapata, 2008] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Comput. Linguist.* 34, 1 (March 2008), 1-34.
- [Galitsky, 2017] Boris Galitsky. Discovering Rhetorical Agreement between a Request and Response. *Dialogue & Discourse* 8(2) 167-205. 2017.
- [Galitsky et al., 2015] Boris Galitsky, D. Ilvovsky, D. and S. Kuznetsov. 2015. Rhetoric Map of an Answer to Compound Queries. *ACL-2*, 681–686.
- [Piwek et al., 2007] Piwek, Paul; Hernault, Hugo; Prendinger, Helmut and Ishizuka, Mitsuru (2007). T2D: Generating Dialogues Between Virtual Agents Automatically from Text. In: *Intelligent Virtual Agents* (Pelachaud, Catherine; Martin, Jean-Claude; Andre, Elisabeth; Chollet, Gerard; Karpouzis, Kostas and Pele, Danielle eds.), *Lecture Notes in Artificial Intelligence*, Springer, Berlin Heidelberg, pp. 161–174.
- [Cox et al., 1999] Cox, R J. McKendree, R. Tobin, J. Lee, and T. Mayes. Vicarious learning from dialogue and discourse: A controlled comparison. *Instructional Science*, 27:431– 458, 1999.
- [Surdeanu et al., 2015] Surdeanu, Mihai, Thomas Hicks, and Marco A. Valenzuela-Escarcega. Two Practical Rhetorical Structure Theory Parsers. *NAACL HLT*, 2015.

mhirano at the FinSBD Task: Pointwise Prediction Based on Multi-layer Perceptron for Sentence Boundary Detection

Masanori HIRANO^{1*}, Hiroki SAKAJI¹, Kiyoshi IZUMI¹, Hiroyasu MATSUSHIMA¹

¹School of Engineering, The University of Tokyo

hirano@g.ecc.u-tokyo.ac.jp, {sakaji, izumi, matsushima}@sys.t.u-tokyo.ac.jp

Abstract

This paper proposes a pointwise prediction for a sentence boundary detection task. The proposed pointwise prediction is combined with our original word embedding method and three-layered perceptron. It predicts whether the targeted words have the role of the beginning/end of a sentence or not by using word features around the targeted words. We tested our model by changing some parameters in our model and then ensembled these models with various parameters. Consequently, the ensembled model achieved 0.88 and 0.84 averaged f1-score by testing the data both in English and French, and it also obtained 0.84 in English and 0.86 in French as the final results of this shared task. In addition, we developed a baseline model, that is, a rule-based prediction model, for comparison. The result shows that the proposed pointwise prediction model outperformed the rule-based prediction model in any index.

1 Introduction

This paper presents the application technique¹ of the pointwise prediction to a shared task of Sentence Boundary Detection in PDF Noisy Text in the Financial Domain for the FinSBD 2019 shared task [Ait Azzi *et al.*, 2019]².

We address the sentence boundary detection problem in PDF Noisy Text using a type of approach referred to as “pointwise” prediction. Pointwise prediction is an approach used to make every single independent decision at each point by using only the features around a single point. In this task, a pointwise prediction indicates predicting whether each word has a role as the beginning/end of the sentence or not by using only the features around that word. This type of approach was also used in Japanese morphological analysis, which is a task to detect boundaries of the smallest meaningful units because Japanese text has no spacing between each word.

*Contact author; <https://mhirano.jp/>

¹Our code is available on <https://s.mhirano.jp/FinSBD2019>

²<https://sites.google.com/nlg.csie.ntu.edu.tw/finnlp/shared-task-finsbd>

The pointwise prediction for Japanese morphological analysis [Neubig and Mori, 2010; Neubig *et al.*, 2011]³ achieved high results. The advantages of this pointwise prediction are its robustness and adaptiveness. Presently, many prediction techniques based on machine learning exist. Machine learning, specifically recurrent neural network (RNN), remarkably depends on features, and when one of the features is wrong, the results through these machine learning can also be wrong. However, the effect of the wrong feature or missing feature information is supposed to be local when this type of pointwise prediction is used. In addition, prediction using machine learning techniques apart from the pointwise prediction, particularly RNN, requires many training data, but the training data in this task are limited. Nevertheless, by using the pointwise prediction, the training data become more abundant than that of other machine learning techniques because the number of sentences is limited and much less than the number of candidates for sentence boundaries. We supposed that the pointwise prediction has some advantages on this prediction task, that is, sentence boundary detection with noise.

For this shared task, we submitted two test predictions, one of which is the result of our pointwise model and the other one is that of a simple rule-based model only. Here, we focused on describing the first one, and the latter is treated as a baseline model. In addition, we used the script that was distributed by the organizer of this shared task for evaluations.

2 Pointwise Prediction Model

First, we explain our pointwise prediction model. Figure 1 shows the model outline.

The proposed prediction model uses words around the target point, which is to be classified into the beginning of a sentence, the end of a sentence, and others. That is, by using a window size N_W , our prediction model utilizes $(N_W \times 2 + 1)$ words, including a target point word and N_W words before and after the target point words. In Figure 1, a window size of 4 is employed. These words to be input into a three-layered perceptron also are embedded into vectors.

Next, we describe details about words embedding and the multi-layer perceptron.

³This Japanese morphological analyzer is called “KyTea.”

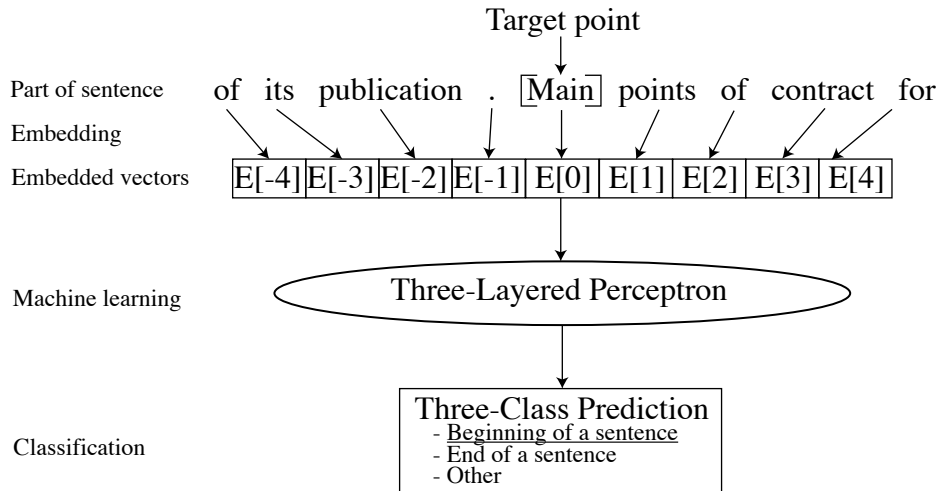


Figure 1: Pointwise prediction outline. This example employs a window size of 4. Words around the target point are gathered and embedded. The embedded vectors are going through a three-layered perceptron and making a prediction for a target point. The prediction has three classes: the beginning of the sentence, the end of the sentence, and other.

2.1 Words Embedding

For words embedding, we use three types of factors. Here are the features for word embedding:

1. Word2vec (gensim in python3)
2. Part-of-speech(POS) tag (NLTK POS tag)
3. CAPITAL/non-capital patterns for each character in a word
4. Whether including line feed code or not

Word2vec [Mikolov *et al.*, 2013] is a method of translating words into vectors. In this task, we used the training data from this shared task organizers for the word2vec training data, and the word2vec window size is set equal to the window size of our model N_W . Moreover, the word embedding dimension of word2vec N_D is set as a hyperparameter. We employed the vectors through this word2vec as the first N_D factors in the embedded vectors.

The POS tag is fetched from the Natural Language Toolkit (NLTK) [Bird, 2006]. This NLTK’s POS tag data is obtained via `nltk` in `python3` and contains 45 types of POS patterns. In case the POS tag classifications in `nltk` fail, we added one additional class to the original 45 classes from NLTK. Hence, this 46-class data are translated into one-hot vectors and used as the subsequent 46 factors in the embedded vectors.

The CAPITAL/non-capital patterns for each character in a word are for recognizing words’ appearance patterns, such as “This,” “this,” “THIS,” and others, including numbers and symbols. We categorized each word into the following patterns:

1. Numbers and/or characters (e.g., 2, #, -, .).
2. All characters are alphabet and non-capital (e.g., this, have).
3. All characters are alphabet and CAPITAL (e.g., THIS, HAVE).

4. All characters are alphabet and only the first character is CAPITAL (e.g., This, Have).

5. Others that are not classified above.

These five classifications also are translated into one-hot vectors and used as the next five factors of the embedded vectors.

The last one factor of the embedded vectors is whether to include the line feed code or not. In the data, line feed code is “\n.”

The details of the embedded vectors are described previously, and the embedded vectors for each word have a total of $N_D + 46 + 5 + 1 = N_D + 52$ factors. In addition, the embedded vectors of $2N_W + 1$ words are concatenated and input into the subsequent three-layered perceptron; hence, the total input vectors’ length become $(N_D + 52) \times (2N_W + 1)$.

2.2 Three-Layered Perceptron

We employed a three-layered perceptron as the classification model. This three-layered perceptron has one input, one output, and two hidden layers. The input layer has $(N_D + 52) \times (2N_W + 1)$ nodes, both the hidden layers have N_H nodes each, and the output layer has three layers. The input and first hidden layers, the first and second hidden layers, and the second hidden and output layers are fully connected. Additional details are presented in Appendix A.

3 Rule-based Prediction Model (Baseline Model)

The rule-based prediction model that we developed is remarkably simple and is constructed as a baseline. This model has two features definition: namely, features definition for beginnings and endings. The features definition of beginnings are as follows:

1. The first character in a word is CAPITAL, whereas the other characters are non-capital.

- The word is “-” (hyphen).
- The word is “(” (left bracket) and the subsequent of the next word is “)” (right bracket).
- The word is “ ” (backquote).

The features definition of endings are as follows:

- The word ending with “.\n” (a period and a line feed code).
- The word is “.” (a period) and the previous and subsequent words are not digits.
- The word ending with “;” (semi-colon).
- The word ending with “;\n” (a semi-colon and a line feed code).
- The word ending with “:\n” (a colon and a line feed code).
- The word ending with “\n” (a line feed code)⁴.

Then, our rule-based model counts these features for each word.

The next process is deciding where the sentences begin and end based on the numbers of features of begins/ends. Basically, the model supposed that the numbers of features of begins/ends are not zero indicate the possibilities of begins/ends. It finds the pairs of the beginning and ending of the sentence that has a minimum length. If the words that have possibilities of the same type of features comes continuously (e.g., the words with possibilities of the beginning of the sentence come again without the appearance of the words having possibilities of the end of the sentence), then we adopt the word that has the highest number of features as the beginning or ending. The algorithm details are presented in Appendix B.

4 Experiments

We were provided with the data by the organizers of this shared task, containing the “training data” and “development data” for two languages, i.e., English and French. We performed the same experiments on each language. In addition, we treated the “training data” as training data and the “development data” as test data. (Actually, the organizers provided also the “test data” to form the leaderboard of this shared task, but we ignored these data in this paper other than final results.) These data contain sentences from the PDF Noisy Text but were split well for each word, symbols, or something, the list of the beginning of the sentences, and the list of the beginning of the sentences⁵.

Using these data, we tested our model. In our model, several unfixed hyperparameters are the following:

- N_W : window size;
- N_D : the dimension for word2vec; and
- N_H : the number of nodes on every single hidden layer in the three-layered perceptron.

⁴This feature sometimes overlaps with other features, but when it is, we count it redundantly.

⁵Details are shown in <https://sites.google.com/nlg.csie.ntu.edu.tw/finnlp/shared-task-finsbd> or [Ait Azzi *et al.*, 2019].

We modified these hyperparameters as in Table 1 and tested the model.

Hyperparameter	Candidates
N_W	5, 10, 200
N_D	10, 50, 100
N_H	300, 600, 1200

Table 1: Hyperparameter candidates for each hyperparameter.

Apart from these parameter sets, we ensembled the results of all models. In the ensembling process, only the words that two-thirds of all models agree with become the beginning/ending of the sentences.

5 Results

N_W	N_D	N_H	BS	ES	Ave.
Ensemble			0.84	0.92	0.88
5	10	1200	0.83	0.92	0.88
10	100	600	0.82	0.91	0.87
5	50	600	0.82	0.91	0.87
10	50	1200	0.81	0.91	0.86
10	100	300	0.81	0.91	0.86
5	10	600	0.81	0.91	0.86
10	50	300	0.81	0.91	0.86
10	10	1200	0.82	0.90	0.86
5	100	1200	0.82	0.90	0.86
5	50	1200	0.82	0.90	0.86
20	50	1200	0.80	0.91	0.86
5	100	600	0.81	0.90	0.86
10	10	600	0.81	0.90	0.86
10	100	1200	0.81	0.90	0.86
20	100	600	0.80	0.91	0.86
5	10	300	0.81	0.90	0.86
20	100	1200	0.80	0.90	0.85
10	10	300	0.81	0.89	0.85
20	10	1200	0.79	0.91	0.85
20	10	600	0.79	0.91	0.85
5	50	300	0.80	0.89	0.85
20	50	600	0.79	0.90	0.85
5	100	300	0.80	0.89	0.85
20	50	300	0.78	0.90	0.84
20	100	300	0.78	0.90	0.84
20	10	300	0.77	0.89	0.83
10	50	600	0.74	0.80	0.77

Table 2: Results of the English language using the pointwise prediction model and its hyperparameter sets. Here, BS/ES indicates f1-score for the prediction of the beginning/ending of the sentences and Ave. denotes the average of BS and ES.

Each result for each parameter sets using pointwise prediction model are shown in Tables 2 and 4. Tables 2 and 3 are results for English and Tables 4 and 5 are results for French. An evaluation tool is provided by the organizers of this shared task. The tool calculates the f1-scores for the prediction of

Ensemble model (En)	Precision	Recall	F1-score
Others	1.00	0.99	0.99
Beginnings	0.81	0.87	0.84
Ends	0.87	0.98	0.92
Micro avg	0.99	0.99	0.99
Macro avg	0.89	0.95	0.92
Weighted avg	0.99	0.99	0.99

Table 3: Detailed results for English using the pointwise prediction ensemble model.

N_W	N_D	N_H	BS	ES	Ave.
Ensemble			0.80	0.88	0.84
10	10	1200	0.79	0.86	0.83
10	50	1200	0.78	0.86	0.82
10	100	1200	0.77	0.87	0.82
10	100	600	0.77	0.86	0.82
20	50	1200	0.75	0.86	0.81
10	10	600	0.76	0.85	0.81
10	50	600	0.77	0.85	0.81
20	100	1200	0.76	0.86	0.81
5	10	1200	0.74	0.85	0.80
20	10	1200	0.73	0.86	0.80
10	50	300	0.73	0.84	0.79
20	10	600	0.73	0.84	0.79
20	50	600	0.74	0.84	0.79
20	100	600	0.74	0.84	0.79
10	10	300	0.72	0.84	0.78
5	10	600	0.72	0.83	0.78
10	100	300	0.72	0.83	0.78
20	100	300	0.71	0.83	0.77
5	100	600	0.70	0.83	0.77
5	100	1200	0.71	0.83	0.77
5	50	1200	0.72	0.82	0.77
5	50	600	0.72	0.82	0.77
20	50	300	0.69	0.82	0.76
5	10	300	0.70	0.82	0.76
5	50	300	0.69	0.80	0.75
5	100	300	0.67	0.80	0.74
20	10	300	0.66	0.81	0.74

Table 4: Results of the French language using the pointwise prediction model and its each hyperparameter sets. Here, BS/ES indicates f1-score for the prediction of the beginning/ending of the sentences and Ave. denotes the average of BS and ES.

Ensemble model (Fr)	Precision	Recall	F1-score
Others	0.99	0.99	0.99
Beginnings	0.74	0.86	0.80
Ends	0.82	0.94	0.88
Micro avg	0.98	0.98	0.98
Macro avg	0.85	0.93	0.89
Weighted avg	0.98	0.98	0.98

Table 5: Detailed results for French using the pointwise prediction ensemble model.

	BS	ES	Ave.
Pointwise Prediction (English)	0.84	0.92	0.88
Baseline: Rule-based (English)	0.73	0.81	0.76
Pointwise Prediction (French)	0.80	0.88	0.84
Baseline: Rule-based (French)	0.67	0.69	0.68

Table 6: Comparing the results of pointwise prediction with ensembling and the results of the baseline model, i.e., rule-based model.

	Test	Final result
Pointwise Prediction (English)	0.88	0.84
Baseline: Rule-based (English)	0.76	0.63
Pointwise Prediction (French)	0.84	0.86
Baseline: Rule-based (French)	0.68	0.68

Table 7: Test and final results of all models and languages. The final result is the result of this shared task and was on the leaderboard.

the beginning/ending of the sentences and others. F1-score denotes the harmonic average of precision and recall.

Table 6 shows the comparison between the results of the pointwise prediction model with ensembling and the results of the baseline model, i.e., rule-based model. In both English and French, our pointwise prediction model outperformed our baseline model in any index.

Table 7 shows the test results from Table 6 and the final results from this shared task’s leaderboard. Predictions for the French language in the test and final results have a slight difference, whereas those for the English language have significant gaps. Specifically, the rule-based model in English performs worse in the final results.

6 Discussion

First, based on Table 6, the proposed pointwise prediction model outperformed the rule-based prediction model. Evidently, the reason for the accurate predictions is that our pointwise prediction model employs a type of feature learning. Rules for the rule-based model were implemented by us and the rules are not sufficient. By using the feature learning, these processes, such as implementing rules, are not necessary and render the model more adaptive for wide cases. We employed only a simple three-layered perceptron, but this model worked well, as observed in the final results’ leaderboard.

Second, the result of pointwise predictions using various parameters suggests interesting insights. Models with $(N_W, N_D, N_H) = (10, 100, 600)$, $(5, 10, 1200)$, and $(5, 50, 600)$ show the highest results in both English and French. Table 8 shows the top results in various parameters. As observed in this table, $N_W = 20$ does not appear. Thus, the window sizes are limited when the results are high. This result shows that the beginning/ending of sentences can be recognized only by watching approximately 5–10 words around. In addition, the size of the hidden layer of the three-layered perceptron N_H of the top results tends to be high.

N_W	N_D	N_H	En	Fr	Ave.
10	100	600	0.87	0.83	0.85
5	10	1200	0.88	0.82	0.85
5	50	600	0.87	0.82	0.85
10	50	300	0.86	0.82	0.84
10	50	1200	0.86	0.81	0.84
10	100	1200	0.86	0.81	0.84
5	50	1200	0.86	0.81	0.84

Table 8: Top of the results in various parameters. Here, Ave. denotes the average of the results of both English and French.

However, when $N_W = 10, N_D = 100$, the input length for the three-layered perceptron must be 3192. Only when $N_W = 5, N_D = 10$, it must be 682. These results suggest that additional hidden layer sizes are possibly necessary

Third, the final result predictions in English were noticeably worse than those of the test results, whereas the French data were not. We assume that the English test data for the final results include some out-of-sample data. Both the pointwise prediction and rule-based models perform poorly; the characteristics of the data might be the cause of these lousy results.

As future works, we have to test additional parameters for N_W, N_D, N_H . Moreover, we must change the fixed parameters in Appendix A. Not only the parameters but also the feature learning models, apart from the three-layered perceptron, should be evaluated for their accuracy.

7 Conclusion

In this paper, we presented the application approach of pointwise prediction to sentence boundary detection in the PDF Noisy Text in the financial domain for the FinSBD 2019 shared task. Our point prediction model achieved 0.88 and 0.84 averaged f1-score for the beginning/ending of sentences in English and French. In the final results, this model obtained 0.84 in English and 0.86 in French. Evidently, the proposed pointwise prediction model outperformed the rule-based prediction model in any index. In our model, we employed some sets of parameters and ensembled models with these parameter sets. The result shows that the ensemble models outperformed any model without ensembling. However, other parameter sets that are also accurate for this task are possible. Moreover, we fixed some parameters. As future works, these parameters should also be modified.

Acknowledgments

Funding from Daiwa Securities Group is gratefully acknowledged.

A Details of the Three-Layered Perceptron

Table 9 shows the parameters of the employed three-layered perceptron. These parameters are totally fixed and not optimized. Therefore, we will enhance these parameters in the future. As observed in table 9, the maximum training epoch is 10,000. However, training was forced to be stopped using Algorithm 1.

	Parameter
Activation function	relu
Loss function	Cross entropy loss
Optimize function	SGD
Dropout	True
Dropuout rate	0.2
Learning rate	0.01
Momentun	0.9
Weight decay	5×10^{-4}
Training data split	90% for training, 10% for evaluating.
Batch size	1,000
Max training epochs	10,000

Table 9: Parameters for three-layered perceptron

Algorithm 1 The algorithm for stop training

Input: Training data

Output: The best performed model

```

1:  $step \leftarrow 0$ .
2:  $max\_acc \leftarrow 0, max\_step \leftarrow 0, acc\_list \leftarrow []$ .
3: loop
4:   Train(90% of training data).
5:    $acc \leftarrow$  Evaluate(10% of training data).
6:    $acc\_list[step] \leftarrow acc$ .
7:   if  $acc \geq max\_acc$  then
8:      $max\_acc \leftarrow acc$ .
9:      $max\_step \leftarrow step$ .
10:    SaveModel().
11:   end if
12:   if  $step \bmod 10 = 0$  and  $step \geq 100$  then
13:     Calculate all 100 steps moving average of  $acc\_list$ .
14:      $M \leftarrow$  (Step with the best moving average).
15:     if  $Step > \max(M \times 1.1, M + 200)$  then
16:       Break.
17:     end if
18:   end if
19:    $step \leftarrow step + 1$ .
20: end loop
21: return  $max\_acc, max\_step$ 

```

B Detailed Algorithm of the Rule-based Prediction Model

The algorithm is shown in Algorithm 2.

Algorithm 2 The algorithm for rule-based prediction

Input: The list of the numbers of features of the beginning/end of the sentences fob, foe

Output: The list of the beginning bos and the list of the end of the sentences eos

```
1:  $i \leftarrow 0$ .
2:  $bos \leftarrow []$ ,  $eos \leftarrow []$ ,  $status \leftarrow False$ .
3:  $last\_start \leftarrow 0$ ,  $last\_end \leftarrow 0$ .
4: loop
5:   if  $fob[i] = 0 \wedge foe[i] = 0$  then
6:     Go to 34.
7:   else if  $fob[i] = 0 \wedge foe[i] \neq 0$  then
8:     if  $status$  then
9:       Append  $last\_start$  to  $bos$ .
10:       $status \leftarrow False$ ,  $last\_end \leftarrow i$ .
11:     else if  $foe[last\_end] < foe[i]$  then
12:        $last\_end \leftarrow i$ .
13:     end if
14:   else if  $fob[i] \neq 0 \wedge foe[i] = 0$  then
15:     if  $status$  then
16:       if  $fob[last\_start] < fob[i]$  then
17:          $last\_start \leftarrow i$ .
18:       end if
19:     else
20:       Append  $last\_end$  to  $eos$ .
21:        $status \leftarrow True$ ,  $last\_start \leftarrow i$ .
22:     end if
23:   else
24:     if  $status$  then
25:       Append  $last\_start$  to  $bos$ .
26:       Append  $i$  to  $eos$ .
27:        $last\_start \leftarrow i$ .
28:     else
29:       Append  $i$  to  $bos$ .
30:       Append  $last\_end$  to  $eos$ .
31:        $last\_end \leftarrow i$ .
32:     end if
33:   end if
34:    $i \leftarrow i + 1$ .
35: end loop
36: return  $bos, eos$ 
```

References

- [Ait Azzi *et al.*, 2019] Abderrahim Ait Azzi, Houda Bouamor, and Sira Ferradans. The FinSBD-2019 Shared Task: Sentence boundary detection in PDF Noisy text in the Financial Domain. In *The First Workshop on Financial Technology and Natural Language Processing (FinNLP 2019)*, Macao, China, 2019.
- [Bird, 2006] Steven Bird. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72, Morristown, NJ, USA, 2006. Association for Computational Linguistics.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of Twenty-seventh Conference on Neural Information Processing Systems (NeurIPS 2013)*, pages 3111–3119, Stateline, Nevada, USA, 2013.
- [Neubig and Mori, 2010] Graham Neubig and Shinsuke Mori. Word-based Partial Annotation for Efficient Corpus Construction. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pages 2723–2727, Valletta, Malta, 2010. European Language Resources Association.
- [Neubig *et al.*, 2011] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011)*, pages 529–533, Portland, Oregon, USA, 2011.

NUIG at the FinSBD Task: Sentence Boundary Detection for Noisy Financial PDFs in English and French

Tobias Daudert* and Sina Ahmadi

Insight Centre for Data Analytics
National University of Ireland, Galway
{tobias.daudert, sina.ahmadi}@insight-centre.org

Abstract

Portable Document Format (PDF) has become the industry-standard document as it is independent of the software, hardware or operating system. Publicly listed companies annually publish a variety of reports and too take advantage of PDF. This leads to the rise in PDF containing valuable financial information and the demand for approaches able to accurately extract this data. Analyzing and mining information requires a challenging extraction phase, particularly with respect to document structure. In this paper, we describe a sentence boundary detection approach capable of extracting complete sentences from unstructured lists of tokens. Our approach is based on the application of a language model and sequence classifier for both the English and the French language. The results show a good performance, achieving F1 scores of 0.855 and 0.91, and placed our team in 3rd and 5th for the French and English language, respectively.

1 Introduction

At a time we face an information deluge, automated solutions tailored to different formats are crucial for the data interpretation. In industry, Portable Document Format (PDF) has become the standard document as it is independent of the software, hardware or operating system in use [Document-CloudTeam, 2013]. Publicly listed companies annually publish a variety of reports and too take advantage of PDF. In addition to factual information and numerical data, such documents provide deeper knowledge which is conveyed through wording and linguistic structure [Thomas, 1997]. With the rise in PDF containing valuable financial information, the demand for approaches able to accurately extract this data is also growing. However, analyzing and mining information requires a challenging extraction phase reliant on the document structure. Sentence boundary detection is vital to understand the document structure. Hence, this is the focus of the FinSB task and this paper.

Although not considered one of the grand challenges in natural language processing (NLP), sentence boundary detec-

tion remains challenging particularly due to textual variation [Read *et al.*, 2012]. Sentence boundary detection (SBD) aims at determining where a sentence begins and ends, in detail, it is the task of binary classifying text into boundary point or non-boundary point after each character [Read *et al.*, 2012]. SBD plays an important role in structuring textual data. For example, machine translation needs correct sentence segmentation as it heavily impacts the translation performance [Walker *et al.*, 2001], and speech recognition requires segmented sentences for the processing in downstream tasks as well as to improve human readability [Liu *et al.*, 2005]. SBD is paramount for text extraction in PDF since a major "problem in the conversion of PDF documents is the detection of the boundaries of common textual units such as paragraphs, sentences and words" [Tiedemann, 2014]. Although SBD is being researched for almost 20 years, the majority of works focus on structured texts (e.g. WSJ corpus, Brown corpus) and little attention is given to SBD in PDFs. In particular, research dealing with sentence boundary detection in financial PDFs is non-existing, to the best of our knowledge. The only related work found was the paper by Loughran and McDonald which deal with the readability of 10-k reports, however, the authors do not target sentence boundaries in their FOG index [Loughran and McDonald, 2014].

In this paper, we define SBD as the ternary classification of a token to identify the *sentence beginning*, *sentence end*, and *other token*. Below, we outline that *other token* variations occur in the form of *in-sentence-token* or *out-of-sentence-token*. Thus, our classification goes a step further and does not only aim at boundary points (i.e. sentence beginning and end) but is also able to determine a sentence within a list of tokens from its beginning to its end. This becomes particularly important for cases in which a sentence does not follow another sentence (e.g. a headline followed by a sentence).

The paper is organized as follows: First, we present work related to this paper; second, we define the research problem, third, we explain our methodology to deal with sentence boundary detection for domain-specific texts in the English and French language; fourth, we present the results of the methodology application and analyze these; lastly, we conclude this work with a methodology and findings summary.

*Contact Author

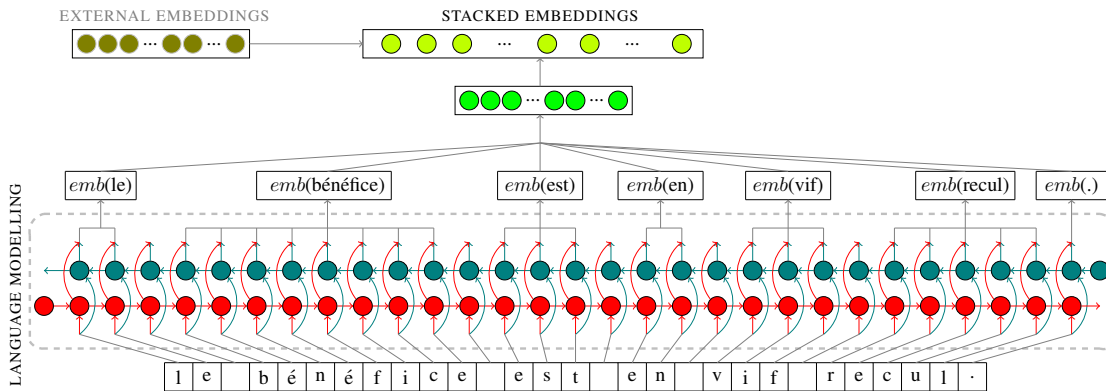


Figure 1: An illustration of our language modelling architecture. A bidirectional recurrent neural network, forward in red and backward in green, with LSTM mechanism retrieves the contextual embedding of each word at character-level. The produced embedding is then merged with an external embedding to create the stacked embeddings.

2 Related Work

Sentence boundary detection is a fundamental preprocessing step for the use of text in downstream tasks such as part-of-speech-tagging and machine translation. While rule-based approaches are the earliest method applied, we focus the related work on more advanced approaches, namely neural networks. The use of neural networks (NN) for sentence boundary detection dates as far back as 1994 [Cutting *et al.*, 1992].

Palmer and Hearst used a NN with two hidden units as an adaptable approach to overcome the restrictions of rule-based sentence boundary detection [Palmer and Hearst, 1994]. Their work utilised the part-of-speech (POS) surrounding sentence endings as an indicator. Since most POS tagger require available sentence boundaries, they inferred the POS based on the previous part-of-speech. When applied on a corpus of Wall Street Journal articles (WSJ), their work correctly disambiguated over 98.5% of sentence ending punctuation marks. Riley uses a decision-tree based approach to detect endings of sentences in the Brown corpus [Riley, 1989]. The maximum entropy approach by Reynar and Ratnaparkhi achieves an accuracy of 98.0 % on the Brown corpus and 97.5% on the WSJ corpus [Reynar and Ratnaparkhi, 1997]. In an effort to segment sentences in the output of vocabulary-speech-recognizers, Stolcke and Shriberg use a statistical language model to retrieve the probabilities of sentence endings [Stolcke and Shriberg, 1996]. They also mention the beneficial impact POS use can have. In a later work, Storcke *et al.* used decision trees to model a combination of prosodic cues aiming at the detection of events (i.e. sentence boundaries and disfluencies) [Stolcke *et al.*, 1998]. Dealing with a similar problem, Gotoh and Renals utilise n-gram language models to predict sentence boundaries from broadcast transcripts which have been converted to text [Gotoh and Renals, 2000]. Stevenson and Gaizauskas approach sentence boundary detection in automated speech recognition transcripts using a memory-based learning approach [Stevenson and Gaizauskas, 2000]. Other works used Hidden Markov Models (HMM) [Shriberg *et al.*, 2000] and Conditional Random Fields (CRF) [Liu *et al.*, 2005; Liu *et al.*, 2006]. Also in a machine translation setting,

sentence boundary detection is important since it affects the translation quality. Walker *et al.* explore the use of three different algorithms to detect sentence boundary as pre-requisite of machine translation [Walker *et al.*, 2001]. They name the first algorithm, which is based on the Barcelona engine as part of the Power Translator, *The Direct Model*. The second algorithm is based on rules and employed as independent preprocessing, contrary to the first algorithm. The third algorithm is essentially a re-implementation of [Reynar and Ratnaparkhi, 1997]. Besides their results, which show the highest performance for the third algorithm, they also argue for its use as it is flexible in terms of adaption to other languages, fast in terms of training and delivers results only requiring a small corpus of labelled data, and straightforward in terms of feature selection. Kiss and Strunk propose a language-independent unsupervised SBD algorithm using Dunning’s log-likelihood ratio on a tagged corpus [Kiss and Strunk, 2006]. Shriberg *et al.* compare different evaluation methods for the task of SBD [Liu and Shriberg, 2007]. Tomalin and Woodland compare two types of prosodic feature models for the task of sentence boundary detection [Tomalin and Woodland, 2006]. Particularly, they compare discriminatively trained Gaussian Mixture Models, CART-style decision trees, and task-specific language models with each other. Their results do not show a difference in performance between Gaussian mixture models and CART-style decision trees. An implementation of a multilingual sentence boundary detector is iSentenizer- μ [Wong *et al.*, 2014]. iSentenizer- μ first creates a binary decision tree, the authors call it offline training, on initially provided training data, which is continuously revised by an incremental tree learning algorithm whenever unseen data arrives.

3 Problem Definition

The goal of this shared task is to predict sentence boundaries from a list of words. Data is provided for two languages: English and French. In detail, it is provided a JSON file containing the fields *text*, *begin_sentence*, and *end_sentence*. *text* contains the unsegmented text to be tagged, *begin_sentence* and *end_sentence* contain the indices of the beginning and

Parameter	Language Model	Sequence Classifier
hidden_size	2048	256
nlayers	1	1
mini_batch_size	100	32
epochs	2	100
sequence_length	250	-

Table 1: Parameter selection values for the language models and the sequence classifier training.

end of a sentence, respectively [Ait Azzi *et al.*, 2019]. In addition, a python script which applies two processing methods is given. On one hand, it splits the text into individual tokens using a white-space tokenizer, on the other hand, it creates a list of *O* tags replacing each *O* with a *BS* or *ES* in case its index is contained in the *begin_sentence* or *end_sentence* fields. After applying the python script to the file, we obtain two lists: The first list contains tokens while the second list contains tags. The used tags are [BS, ES, O] with BS representing the beginning of a sentence, ES the end of a sentence, and O other.

4 Methodology

In this section, we describe the approach designed to tackle the problem described in 3. It relies on two pillars: 1) The creation of two language models for each language to use as additional data, and 2) the training of two sequence classifier to tag the test data.

4.1 Language Modelling

One aim of language models is "to learn the joint probability function of sequences of words in a language" [Bengio *et al.*, 2003]. This makes them useful to our task as we can reformulate the sentence boundary detection challenge as a probabilistic problem in which we want to determine whether the following word in a string belongs to a sentence, given all previous words. Furthermore, recent developments in neural-network-based language models have shown relevant improvements, hence, we take advantage of such an approach for the extraction of word embeddings (e.g. [Devlin *et al.*, 2018; Peters *et al.*, 2018]).

The data provided in this shared task consists of PDFs containing financial prospectus, hence, we aimed at identifying corpora providing similar texts (i.e. organizational writing) for the training of the language models. For SBD in English texts, our choice fell on two corpora; the 10-k Corpus which contains 10-k reports filled by US companies between 1996 and 2006 [Kogan *et al.*, 2009], and the JoCo Corpus which consists of annual reports and company responsibility reports of a diverse set of companies (e.g. DJIA, FTSE100, DAX, S&P500, NASDAQ) collected between 2000 and 2015 [Händschke *et al.*, 2018]. Together, both corpora provide us with a diverse set of organizational writing from a period of 20 years in the English language. We further cleaned both corpora using multiple regular expressions to remove irregular breaklines and spacing, HTML tags, and JavaScript strings. Regarding SBD in French texts, we have not been able to locate an appropriate corpus

containing financial texts, especially organizational writing. Therefore, we created a novel corpus containing 2655 company reports, amounting to over 188 million tokens, from the 60 largest French companies by market capitalization, published between 1995 and 2018 [Ahmadi and Daudert, 2019].

The joint English corpus and CoFiF are then used to train two character-level language models using recurrent neural networks (RNN) for each language as illustrated in Figure 1. The language model is composed of two independent RNNs, in the forward direction and the backward direction, shown respectively in red and green in Figure 1. In the forward direction, the input sequence is fed in normal time order while in the backward direction, in reverse time order. The outputs of the two networks are concatenated at each time step. We used the publicly available NLP library Flair [Akbik *et al.*, 2018; Akbik *et al.*, 2019] for the language modelling. The training details are shown in table 1.

We conducted experiments to determine the quality of the trained language models by employing sentence perplexity calculations. The experimental results for the English language models are detailed below; the evaluation of the French language models is described in [Ahmadi and Daudert, 2019]. In both cases, we randomly selected 100 sentences from annual reports external to the corpora and rendered these meaningless by removing or replacing words, or characters. Having 100 correct sentences and 100 incorrect ones in place, we queried the language models for the sentence perplexity score for each sentence. The language model prediction is correct if it provides a lower perplexity score for the original sentence and a higher score for the modified sentence. Three sample sentences are shown in table 2. The language models are then used to extract word embeddings to use in the sequence classifier.

4.2 Sequence Classification

Given the similarity of the stated problem with part of speech (POS) tagging, we choose to re-train a sequence classifier also provided by Flair, as it has shown state-of-the-art performance on POS-tagging [Akbik *et al.*, 2018]. Instead of a list of tuples containing a word and the respective label, our sequence classifier requires segmented sentences. As the input for the sequence classifiers requires a TSV format, segmented sentences are separated by empty lines in the training data. Hence, we preprocess the training, development, and test data inserting an empty line after each $\backslash n$. Furthermore, we conduct a second modification as part of our experiments; this modification includes manipulation of the labels. The originally provided data contains the labels [BS,ES,O] (section 3); we refer to their use as approach 1. However, we aim to provide further information to the classifier by introducing a fourth label [BS,ES,IS,O]. We refer to this as approach 2. The label IS stands for *in-sentence* and is determined during the preprocessing by labeling all words after begin-of-sentence and before end-of-sentence, as IS (i.e. in-sentence). Consider the following text "*October 2013 Distribution of this prospectus is not authorised.*", where *October 2013* is part of the header. Approach 1 would label *Distribution* as $\langle BS \rangle$ and *.* as $\langle ES \rangle$; the remaining tokens are labelled as $\langle O \rangle$. Whereas approach 2 would label *October* and

Sentence	Perplexity
GET SA's shares and the NRS issued by EGP have been listed on the London Stock Exchange since 2 July 2007.	2.9654
GET SA's shares and the NRS issued by EGP have been listed on the London Stock Exchange from 2 July 2007.	3.0157
The board of directors of GET SA has endeavoured to set up appropriate committees as envisaged by its internal procedures.	2.4056
The board of directors of SA has endeavoured to set up appropriate committees as envisaged by its internal procedures.	2.3441
In cooperation with the SNCF, Europorte 2, the rail freight subsidiary of Eurotunnel Group has just started its operational activity in the Frethun cross-Channel rail freight depot adjacent to the French end of the Tunnel.	3.9674
In cooperation with the , Europorte 2, the rail freight subsidiary of Eurotunnel Group has just started its operational activity in the Frethun cross-Channel rail freight depot adjacent to the French end of the Tunnel.	3.8726

Table 2: Six English sample sentences and their perplexity scores according to the character-level forward language model. The upper sentence of each pair is the original sentence, the lower sentence is the modified and wrong sentence.

2013 as $\langle O \rangle$, *Distribution* as $\langle BS \rangle$, $\langle ES \rangle$, and the remaining tokens as $\langle IS \rangle$, since these occur between $\langle BS \rangle$ and $\langle ES \rangle$, to form a valid sentence. Although the ultimate goal is only to predict the BS and ES label, our intuition behind providing this additional knowledge to the classifier is that it might learn to differentiate between sentences and non-sentences (e.g. headlines) as a complete sequence, with BS being at the beginning of a sentence and ES at the end.

To fine-tune the sequence classifier parameters we split the development data into a development set and a test set by the ratio 70% / 30%; having a temporary test set available before the actual test data is released allowed us to experiment with the classifier. When the test data was released, only the last three sentences of the development data were used to test the final classifier and the remaining were included in the training set.

To train the classifier, the first step is to vectorize the data; to achieve this, we use the concept of stacked embeddings [Ammar *et al.*, 2016] and the embeddings from our language models. For the English data, we stack GloVe embeddings [Pennington *et al.*, 2014] with embeddings from the forward language model, and embeddings from the backward language model. Whereas for the French data, we stack fastText embeddings [Grave *et al.*, 2018] with embeddings from the forward language model, and embeddings from the backward language model [Ahmadi and Daudert, 2019]. As the pre-trained GloVe embeddings are only available in English, fastText was chosen in the French data vectorization. With the data prepared, the sequence classifiers were trained for each of the approaches and languages; the training parameters are presented in table 1.

5 Results

The results achieved are presented in two parts: we first provide an analysis of the created language models, and then report on the sequence classifier performance.

5.1 Language Model Evaluation

To evaluate the language model quality, we employed the sentence perplexity-based approach described in section 4.1. Although the sentence perplexity is not used directly to refine the sequence classifiers output, it influences the quality of the stacked embeddings which we employ to train the sequence classifiers. Thus, a good quality language model is imperative for the classification output. The character-level forward language model was tested on 117 random sentences extracted from an additional annual report. The model correctly identified 102 as original sentences and failed to detect 15; three sentence pair examples are shown in table 2, the top sentence is the original/correct version. A lower sentence perplexity score indicates a higher probability for the sentence to appear in this form. Considering these examples, in the first pair, we replaced *since* with *from* which rendered the sentence grammatically incorrect. The difficulty in the second example consists in knowing the structure of French company names, specifically that SA stands for *Société anonyme*, a company type; with the removal of *GET* the model failed to capture that this string/part of the company name is missing. However, we need to keep in mind that the English training data did not contain reports of French companies, thus, it is unlikely our language model has come across such names before. In the third example, we removed the company name *SNCF*. Although this mistake seems obvious to a human, the language model did not detect it. Looking closely at the wrong sentence, one can also understand it as "*In cooperation with the Europorte 2, the rail [...]*" and, hence, only see a misplaced comma.

5.2 Sequence Classification

The sequence classifiers are evaluated with the F1 score for the sentence boundary labels [BS, ES]. The results are shown in table 3. For the shared task in English, our approaches rank 5th and 12th out of 18 submissions; for the French task, our approaches rank 3rd and 4th out of 15 submissions. Comparing both approaches, the results for French and English are the same or higher in approach 1 than approach 2. This indi-

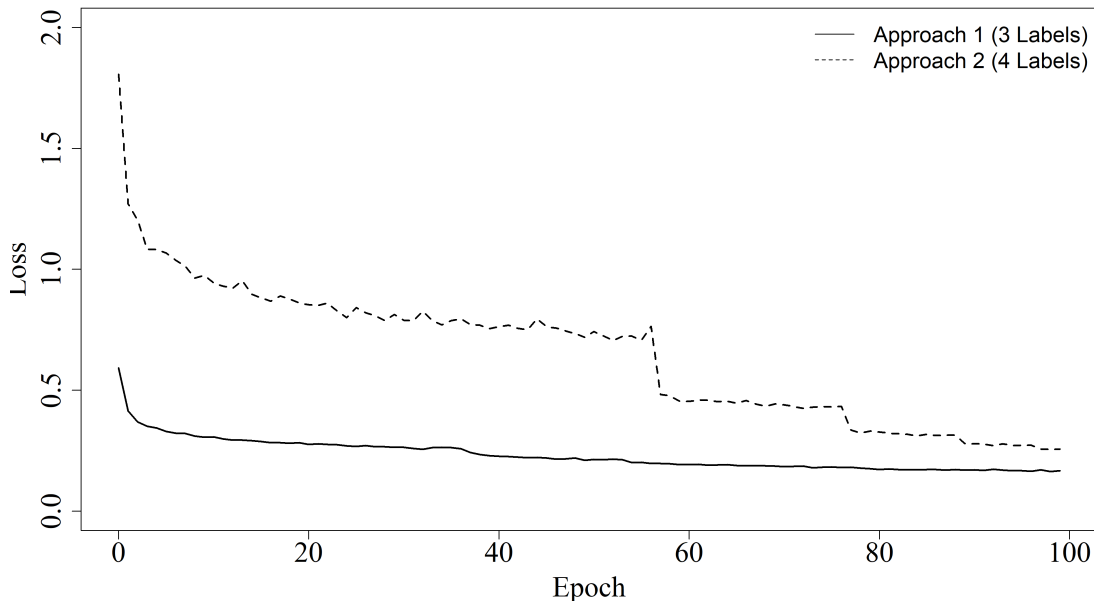


Figure 2: Training loss per epoch during the English sequence classifier training.

cates that the addition of a fourth label (IS) did not improve the classification. Figure 2 shows the training loss during the English classifier training. While the training loss for approach 1 steadily decreases, the decrease for approach 2 is rather unsteady. This volatility could also be an indicator for the difficulty in training the sequence classifier on data containing 4 labels. For the French data, the loss behaviour is similar.

The bottom graph in figure 2 shows a stabilizing loss towards the end of the training. This suggests that a prolonged training period could yield improved results. Nonetheless, a 4-label classification is inherently more difficult than a 3-label classification. The nuances between $\langle O \rangle$ and $\langle IS \rangle$ also suggest that additional training data is required.

Additionally, although we employ the same approaches to both languages, the F1 scores for French are generally higher than for English. We hypothesize this is due to three reasons: 1) the French language models are trained on data more similar to the test data, hence, providing better word embeddings for this particular task; 2) the stacked embeddings using fastText provide a better generalization than stacking embeddings with GloVe; 3) French financial reports structure is stricter, making the sentence boundaries more predictable. We also have to consider natural language differences between English and French which can have an effect on the performance of classification tasks. For all languages and approaches, the F1 scores for the end-of-sentence tag are higher than for the begin-of-sentence tag. We can also observe that approaches 1 and 2 achieve the same F1 score on French while approach 1 achieves different F1 scores for the end-of-sentence tag for English.

Language	Approach	F1 score		Mean F1 score
		BS	ES	
English	1 (3-labels)	0.81	0.9	0.855
	2 (4-labels)	0.81	0.85	0.83
French	1 (3-labels)	0.9	0.92	0.91
	2 (4-labels)	0.9	0.92	0.91

Table 3: sequence classifier evaluation results. The BS and ES tag represent begin-sentence and end-sentence.

6 Conclusions

In this paper, we described our approach to detect sentence boundaries in a corpus of unsegmented text. This approach is tested on English and French data. To this purpose, we utilize two powerful character-level language models, as well as two sequence classifiers for each language. In addition, we target two approaches, one based on the original labels and another introducing a modified label set. Our results yield a good performance placing us at the 3rd rank of this shared task for French and 5th for English. Specifically, the submitted approach for French achieves an F1-score of 0.91 while the approach for English retrieves an F1 score of 0.855.

Our results suggest that fine-tuning the models by training two domain-specific language models and using these to retrieve word embeddings as input for the sequence classifier is key to the achieved performance. Furthermore, we believe that the use of embeddings from other domains (i.e. GloVe and fastText) also contributed to the performance as it avoids a narrow domain focus.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, co-funded by the European Regional Development Fund.

References

- [Ahmadi and Daudert, 2019] Sina Ahmadi and Tobias Daudert. Cofif: A corpus of financial reports in french language. In *The First Workshop on Financial Technology and Natural Language Processing (FinNLP 2019)*, Macao, China, 2019.
- [Ait Azzi *et al.*, 2019] Abderrahim Ait Azzi, Houda Bouamor, and Sira Ferradans. The finsbd-2019 shared task: Sentence boundary detection in pdf noisy text in the financial domain. In *The First Workshop on Financial Technology and Natural Language Processing (FinNLP 2019)*, Macao, China, 2019.
- [Akbik *et al.*, 2018] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.
- [Akbik *et al.*, 2019] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Pooled contextualized embeddings for named entity recognition. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, page to appear, 2019.
- [Ammar *et al.*, 2016] Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*, 2016.
- [Bengio *et al.*, 2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.
- [Cutting *et al.*, 1992] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Third Conference on Applied Natural Language Processing*, 1992.
- [Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [DocumentCloudTeam, 2013] DocumentCloudTeam. Top 10 reasons to use pdf instead of word, excel or powerpoint, 2013.
- [Gotoh and Renals, 2000] Yoshihiko Gotoh and Steve Renals. Sentence boundary detection in broadcast speech transcripts. In *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.
- [Grave *et al.*, 2018] Edouard Grave, Piotr Bojanowski, Prakhara Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [Händschke *et al.*, 2018] Sebastian GM Händschke, Sven Buechel, Jan Goldenstein, Philipp Poschmann, Tinghui Duan, Peter Walgenbach, and Udo Hahn. A corpus of corporate annual and social responsibility reports: 280 million tokens of balanced organizational writing. In *Proceedings of the First Workshop on Economics and Natural Language Processing*, pages 20–31, 2018.
- [Kiss and Strunk, 2006] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.
- [Kogan *et al.*, 2009] Shimon Kogan, Dmitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280. Association for Computational Linguistics, 2009.
- [Liu and Shriberg, 2007] Yang Liu and Elizabeth Shriberg. Comparing evaluation metrics for sentence boundary detection. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–185. IEEE, 2007.
- [Liu *et al.*, 2005] Yang Liu, Andreas Stolcke, Elizabeth Shriberg, and Mary Harper. Using conditional random fields for sentence boundary detection in speech. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 451–458. Association for Computational Linguistics, 2005.
- [Liu *et al.*, 2006] Yang Liu, Nitesh V Chawla, Mary P Harper, Elizabeth Shriberg, and Andreas Stolcke. A study in machine learning from imbalanced data for sentence boundary detection in speech. *Computer Speech & Language*, 20(4):468–494, 2006.
- [Loughran and McDonald, 2014] Tim Loughran and Bill McDonald. Measuring readability in financial disclosures. *The Journal of Finance*, 69(4):1643–1671, 2014.
- [Palmer and Hearst, 1994] David D Palmer and Marti A Hearst. Adaptive sentence boundary disambiguation. In *Proceedings of the fourth conference on Applied natural language processing*, pages 78–83. Association for Computational Linguistics, 1994.
- [Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [Peters *et al.*, 2018] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- [Read *et al.*, 2012] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. Sentence boundary de-

- tection: A long solved problem? *Proceedings of COLING 2012: Posters*, pages 985–994, 2012.
- [Reynar and Ratnaparkhi, 1997] Jeffrey C Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, pages 16–19. Association for Computational Linguistics, 1997.
- [Riley, 1989] Michael D Riley. Some applications of tree-based modelling to speech and language. In *Proceedings of the workshop on Speech and Natural Language*, pages 339–352. Association for Computational Linguistics, 1989.
- [Shriberg *et al.*, 2000] Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tür, and Gökhan Tür. Prosody-based automatic segmentation of speech into sentences and topics. *Speech communication*, 32(1-2):127–154, 2000.
- [Stevenson and Gaizauskas, 2000] Mark Stevenson and Robert Gaizauskas. Experiments on sentence boundary detection. In *Sixth Applied Natural Language Processing Conference*, 2000.
- [Stolcke and Shriberg, 1996] Andreas Stolcke and Elizabeth Shriberg. Automatic linguistic segmentation of conversational speech. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, volume 2, pages 1005–1008. IEEE, 1996.
- [Stolcke *et al.*, 1998] Andreas Stolcke, Elizabeth Shriberg, Rebecca Bates, Mari Ostendorf, Dilek Hakkani, Madeleine Plauche, Gokhan Tur, and Yu Lu. Automatic detection of sentence boundaries and disfluencies based on recognized words. In *Fifth International Conference on Spoken Language Processing*, 1998.
- [Thomas, 1997] Jane Thomas. Discourse in the marketplace: The making of meaning in annual reports. *The Journal of Business Communication (1973)*, 34(1):47–66, 1997.
- [Tiedemann, 2014] Jörg Tiedemann. Improved text extraction from pdf documents for large-scale natural language processing. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 102–112. Springer, 2014.
- [Tomalin and Woodland, 2006] Marcus Tomalin and Philip C Woodland. Discriminatively trained gaussian mixture models for sentence boundary detection. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.
- [Walker *et al.*, 2001] Daniel J Walker, David E Clements, Maki Darwin, and Jan W Amtrup. Sentence boundary detection: A comparison of paradigms for improving mt quality. In *Proceedings of the MT Summit VIII*, volume 58, 2001.
- [Wong *et al.*, 2014] Derek F Wong, Lidia S Chao, and Xiaodong Zeng. isentimizer-: Multilingual sentence boundary detection model. *The Scientific World Journal*, 2014, 2014.

HITS-SBD at the FinSBD Task: Machine Learning vs. Rule-based Sentence Boundary Detection

Mehwish Fatima¹ and Mark-Christoph Müller²

^{1,2}HITS gmbH, Heidelberg, Germany

{mehwish.fatima, mark-christoph.mueller}@h-its.org

Abstract

This paper presents two different approaches towards Sentence Boundary Detection (SBD) that were submitted to the FinSBD-2019 shared task. The first is a supervised machine learning approach which tackled the SBD task as a combination of binary classifications based on TF-IDF representations of context windows. The second approach is unsupervised and rule-based and applies manually created heuristics to automatically annotated input. Since the latter approach yielded better results on the Dev set, we submitted it to evaluation for English and reached an F score of 0.80 and 0.86 for detecting begin of sentences and end of sentences, respectively.

1 Introduction

Sentences are the fundamental units of text, consisting of words and punctuation, and constructing phrases and paragraphs [Reynar and Ratnaparkhi, 1997]. Sentence Boundary Detection (SBD), or finding the start and end of sentences, is an essential prerequisite in the Natural Language Processing (NLP) pipeline for various applications, such as Discourse Parsing [Polanyi *et al.*, 2004], Machine Translation, Document Summarization, Alignment of Parallel Text, Sentiment Analysis, and Information Retrieval [Jonathon *et al.*, 2012]. SBD can have a strong impact on the performance of these applications due to error propagation in the NLP pipeline.

Probably the most important factors in SBD are 1) ambiguous expressions and 2) source of text. In disambiguating sentence boundaries, the most misleading expression is the period (.), which is not only used as end of sentence (ES) marker, but also in abbreviations or numerical expressions (e.g. ordinals and dates). Some other examples of problematic expressions are question mark (?), exclamation mark (!), and colon (:). The other key factor is the genre and / or quality of the text, which can impact the performance of an SBD system. Most of the existing SBD systems are highly accurate on formal and high quality text, but their performance often degrades when the input text is noisy or informal.

The research work for SBD mostly focuses on disambiguating the ends of sentences in formal, noise-free text [Agarwal *et al.*, 2005; Kiss and Strunk, 2006; Akita *et al.*,

2006; Gillick, 2009; Rudrapal *et al.*, 2015]. The FinSBD-2019 Shared Task, in contrast, tackles SBD in noisy text extracted from PDFs from the financial domain.

We applied two different approaches to solve the SBD problem. The first approach treats the problem as an supervised classification task on TF-IDF based representations of context windows, thus taking advantage of the annotated training data set. The second approach is unsupervised, rule-based and applies manually created heuristics to automatically annotated input.

The rest of the paper is structured as follows: Section 2 covers some existing work on SBD. Section 3 explains the general configurations for experiments, including details about the data set and evaluation measures. Sections 4 and 5 explain the two approaches in more detail, and Section 6 concludes the paper.

2 Related Work

This section covers some studies that used rule- or machine learning-based approaches for SBD.

[Agarwal *et al.*, 2005] applied a combination of rule-based features and a probability based classifier (MaxEnt) to predict sentence ending boundaries. They considered three different data sets: 1) Wall Street Journal (WSJ) (43,948 sentences), 2) Penn Treebank (24,243 sentences) and 3) POS-labelled GENIA (20,544 sentences). The feature extraction was executed on each context trigram of data set where each context trigram contained its own label along with a label that either center token is an end of sentence or not. The best results were obtained on WSJ with an F score of 97.8%.

[Kiss and Strunk, 2006] worked on SBD by using different heuristics (ratios, length, etc.) and collocation information for finding abbreviations, initials, and ordinal numbers in an unsupervised manner. The system named ‘Punkt’ was evaluated on news data for eleven different languages and produced a high F score of 91.50% and 97.22% on Wall Street Journal (WSJ) and Lacio Web data sets respectively.

[Akita *et al.*, 2006] experimented with SBD on Japanese spoken transcripts extracted from Corpus of Spontaneous Japanese (CSJ). The total corpus consisted of 200 selected conversations out of which 30 conversations were used as test set. The corpus was annotated both manually and automatically. Two different approaches were applied, a statistical language model (SLM) and Support Vector Machine

(SVM) model. For SLM, they extracted three different feature categories: i) *Linguistic*, which consisted of word surface, reading, POS tags, conjugation based features, and so on, for each context based window, ii) *Pause*, which was calculated on normalized duration (if available), and iii) *Dynamic*, which was extracted by estimating the results of preceding words. For SVM, [Akita *et al.*, 2006] considered the task as text chunking and adopted three categories: i) I (inside), ii) O (outside), and iii) E (end) of chunk, respectively. They applied the YamCha text chunker which is based on SVM with polynomial kernel functions, and achieved an F score for manual annotations of 0.854 and 0.818 for SLM and SVM respectively.

[Gillick, 2009] combined rule-based features and a machine learning approach to SBD. They extracted the rule-based features from training data and fed them into the Support Vector Machine (SVM). For training, WSJ data was used and then the trained model was applied to New York Times data from the AQUAINT corpus to automatically annotate 100 million words. [Gillick, 2009] reported the error score (lowest 0.25%) instead of F score for presenting the results of their system.

[Orosz *et al.*, 2013] presented a hybrid system based on a rule-based approach and unsupervised machine learning for clinical data in Hungarian language. The data set consisted of 1,320 lines in Dev set and 1,310 lines in Test set respectively, without having a train set. They achieved an F score of 91.89% on the test set.

[Rudrapal *et al.*, 2015] collected social media text from Facebook, Twitter and manually annotated the data for sentence breaking utterances. The final corpus was composed of 6,444 sentences in total. [Rudrapal *et al.*, 2015] applied two different approaches, rule-based and machine learning based, to solve the SBD task. Before applying any of approaches, the sentences were tokenized using CMU tokenizer to remove the ambiguities of end of sentence markers. [Rudrapal *et al.*, 2015] obtained an F score of 78.72% and 87.0% with the rule-based approach and SMO classifier respectively.

[Kreuzthaler and Schulz, 2015] worked on abbreviation and SBD with a supervised approach on medical narratives in German language. The authors collected patient discharge summaries from the Graz University Hospital for a period of more than 6 years. The final data set consisted of 1,696 documents which were split into half for generating training and testing set. Different rule-based features were extracted from the data based on length, word information, and contextual and language information. These features were passed to an SVM classifier in the training phase for abbreviations and SBD individually, and achieved the F score of 0.95 and 0.94, respectively.

More recently, [Wiechetek *et al.*, 2019] conducted research on a North Sámi data set consisting of the Uralic language, which has a complex morphological structure and is spoken in Norway, Sweden and Finland. [Wiechetek *et al.*, 2019] applied a context based approach for feature extraction by using constraint grammar and some other structural based information. They report promising results for detecting sentence boundaries with 97% accuracy and 99.99% recall.

Data set	English		French	
	Tokens	Sentences	Tokens	Sentences
Train	904,057	22,342	827,825	22,636
Dev	49,859	1384	119,008	3141
Test	56,952	1265	106,577	2981

Table 1: Statistics of the FinSBD-2019 Data Set

3 Experimental Setup

This section covers the data set and configuration used for our experiments.

3.1 Data set

The Fin-SBD data set [Ait Azzi *et al.*, 2019] was provided in JSON format along with original PDF files. The data set was comprised of three different parts: i) Train set, ii) Dev set, and iii) Test set, for English and French. Table 1 shows the basic data set statistics. Each token was annotated as either *ES* (End of Sentence), *BS* (Begin of Sentence), or *O* (Ordinary token).

3.2 Evaluation Measures

The evaluation metrics include Precision, Recall and F score, which can be automatically computed by the supplied evaluation script for all three labels. Due to the overwhelming number of tokens that are labeled as *O*, the official system performance was only computed as the average F score for *BS* and *ES* label prediction. Also, the standard result format has only two decimal places, i.e. F score ranges from 0.00 to 1.00. While this might be sufficient for overall ranking of shared task participants, we found it too coarse-grained, especially in the detailed analysis of the rule-based system (cf. Section 5), for which we changed the result precision to five decimal places.

4 ML based SBD with Contextual Windows

This section describes our machine learning based SBD approach. It exploits the advantages of supervised machine learning along with context based information for a token passed as a window.

4.1 Generation of Contextual Windows

We utilize the surrounding information of a token as a context window to determine whether this token is a boundary token (either begin or end) or not. A contextual window (*CW*) for a center token at position i consists of some preceding tokens ($i-1, i-2, \dots, i-n$), the token itself (i), and some succeeding tokens ($i+1, i+2, \dots, i+n$) depending on the size of the window (n). A context window (*CW*) of size n can be generalized for a center token T_i as follows:

$$CW_i = T_{i-n} + \dots + T_{i-1} + T_i + T_{i+1} + \dots + T_{i+n}$$

The size of *CW* depends on how much context we want to take into account. For example, a context of size 1 will result in a *CW* size of 3, consisting of one preceding token, the center token, and one succeeding token. We calculated different sizes of *CW*, i.e. 3, 5, 7 and 9. Each *CW* is labeled for center token only, therefore, depending on that token, a *CW* can have one of the three labels *BS*, *ES*, or *O*. A *CW* is

generated for each token that is present in the data set, with majority of the CW s falling into the O category.

4.2 Reduction of Contextual Windows

Generating a CW for each token would result in a huge list of windows that might include duplicates as well. As mentioned earlier, the majority of CW s are labeled O , and it is undesirable to have a highly imbalanced data set for the classification task as it can strongly impact the performance of classifier. Therefore, we opted for two strategies to remove the duplicates and to mitigate the effect of the inherently imbalanced nature of the data set.

Handling Majority Class

To optimize the approach by reducing the majority class O , we applied a selection criterion for each center token to decide whether to add the corresponding CW into the list or not. For application of this criterion, we generated two dictionaries based on the Train set consisting of all unique tokens labeled as BS and ES . We utilized the given BS and ES indices to get all BS and ES tokens from the Train set and then finding unique lists for both. The total number of unique BS and ES tokens are 921 and 216 for English, 790 and 575 for French Train data, respectively. These dictionaries served as selection criteria that if a center token is present in the dictionary, then the corresponding CW will be added in the final CW list, hence resulting in a smaller number of O labeled CW s than before.

Removing Duplicates

To remove the duplicates from the final CW list, at the time of inserting a new CW , a search was made on entire list to check if current CW is already present in the list or not. If current CW is already present, then it will not be added again to handle the redundancy. Finally, each input document was converted into a list of context based windows (CW s) which can be passed to a machine learning classifier for predicting labels for each center token of CW . To preserve the mapping of the original tokens with their indices, we stored the index of each center token for each CW .

4.3 Feature Representation

For applying a machine learning algorithm on textual data, we have to transform the textual data into some numeric or vector representation. There are different methods available for this, such as occurrence based, term-frequency based, TF-IDF (Term Frequency-Inverse Document Frequency), word co-occurrence matrix, and so on. TF-IDF is an occurrence based vector representation of text where TF (Term Frequency) represents the normalized score of word occurrence by the size of the document [Joachims, 1997]. TF of a word w can be expressed as below where D denotes to Document [Joachims, 1997].

$$TF(w) = \frac{Count(w) \text{ in } D}{Total \ w \ \text{ in } D}$$

The result of TF is the assignment of the same weights for each word in the vector representation, which is undesirable. This is because discriminating information of text is mainly

contained in words *other than* stop words, articles, prepositions, etc which occurred a lot in a document. Therefore, such unwanted words should have reduced weights, and the process of suppressing TF scores is done with IDF (Inverse Document Frequency). IDF of a word w can be calculated as follows where D denotes to Document.

$$IDF(w) = Log\left(\frac{Total \ \# \ of \ D}{\# \ of \ D \ \text{ having } w}\right)$$

TF-IDF is the product of both scores which converts the textual data into a vector representation where discriminative words have a higher score than others. We applied TF-IDF vector representation to CW s for generating character N-grams. This resulted in variable length of character N-grams where the minimum length of N is 3 and the maximum length of N is 10. The TF-IDF vector representation usually generates a huge vector space which is computationally expensive. Therefore, we selected the top 5,000 features only.

4.4 Classification

We performed supervised machine learning to exploit the benefit of given annotated data set. To simplify the task, we split the task into two binary classifications: BS vs. O and ES vs. O . For classification, we opted for Random Forest ensemble classifier (decision trees = 100) due to its unbiased and stable nature and Naive Bayes classifier as a baseline. We provided TF-IDF based features on different CW sizes as an input to the classifiers. The Train and Dev data sets were used for training and evaluation to find the optimized configuration. We found the best results from both classifiers with CW size = 5 among all sizes. After that, Train and Test data sets were used for getting final predictions for Test set. Each predicted label was stored against the original index number (cf. Section 4.2), so it could be mapped back into the Test data. The given data set is the full list of tokens and their indices. The indices for which no prediction is made are marked as O by the given evaluation script. The final step is to merge the predictions of both classification tasks to maintain the structure and format of the given data set. As we treated BS and ES detection as separate tasks, so there are two predictions for each token. For each token, we require only one prediction, so two cases arise here for selecting a single prediction for each token, BS/ES vs. O , and BS vs. ES . We resolved this according to the evaluation script where all tokens are initialized with O , then BS markers are placed, and finally ES markers are written.

4.5 Results

Table 2 presents the results of the machine learning based approach with contextual windows CW s of size 5. The first column presents the data set for which the results are reported in the corresponding row. The second column denotes to the classifier used for the classification. The next columns are grouped together to present the results of BS and ES detection in terms of Precision (P), Recall (R) and F score (F). For the English data set, the highest results are obtained on the Dev set for ES (F score = 0.88) and BS (F score = 0.70) detection respectively with the Random Forest classifier, while

Set	Clf	BS			ES		
		P	R	F	P	R	F
EN							
Dev	RF	0.65	0.77	0.70	0.83	0.93	0.88
Test		0.60	0.72	0.65	0.74	0.94	0.83
Dev	NB	0.29	0.61	0.40	0.62	0.77	0.69
Test		0.34	0.60	0.44	0.56	0.76	0.65
FR							
Dev	RF	0.73	0.75	0.74	0.77	0.91	0.84
Test		0.76	0.79	0.77	0.81	0.91	0.85
Dev	NB	0.38	0.57	0.46	0.52	0.78	0.63
Test		0.38	0.56	0.45	0.54	0.77	0.63

Table 2: Results of ML based Approach with *CW* Size = 5

performance of the classifier dropped on the Test set for *ES* (F score = 0.83) and *BS* (F score = 0.65) detection respectively. The same behavior can be observed with the Naive Bayes classifier, i.e. better results are produced on Dev set for *ES* detection (F score = 0.69), however, for *BS* detection (F score = 0.44), the results of Test set are more satisfactory.

Overall, Recall is greater than Precision for Random Forest, which shows that the classifier learned very well despite the imbalanced nature of the data set. For Naive Bayes, Recall is also higher than Precision, however, low Precision results in low F score. For the French data set, the best results are obtained on Test set for *ES* (F score = 0.85) and *BS* (F score = 0.77) detection respectively with Random Forest classifier, and the performance of the classifier was a bit less on Dev set for *ES* (F score = 0.84) and *BS* (F score = 0.74) detection respectively. Interestingly, the performance of the Naive Bayes classifier on Dev and Test sets is very similar for *ES* (F score = 0.63) and *BS* detection (F score = 0.46 for Dev and F score = 0.45 for Test set). A similar pattern is identified for Recall and Precision where Recall is better than Precision for Random Forest, which also confirms the robust nature of the classifier. For Naive Bayes, Recall is also higher than Precision, however, low Precision resulted in low F score.

Comparing the results of the *BS* and *ES* detection task, overall the results of *ES* detection are higher than *BS* detection which depicts that *BS* markers are harder to find than the *ES* markers. The other reason for better *ES* detection result can be correlated with the total number of unique marks. As mentioned in Section 4.2, the total number of unique *ES* markers is much smaller than that of *BS* markers. Probably, the smaller number of unique markers resulted in better learning of classifier. Regarding the languages, the classifiers showed good performance for French on Test set considering both *ES* and *BS* detection.

5 Rule-based SBD

The general strategy underlying our alternative, rule-based approach can be characterized as follows: In an initial step, for easier access at the following stages, the input document is represented as a sequence of numbered tokens which are read from the original JSON data set. We do this by extracting the content of the ‘text’ element from the JSON file and splitting it on the basis of white space, thus maintaining the original tokenization and the gold data token indices, which are stored as the first item in a 4-tuple (cf. Figure 1). Then, we strip any trailing newlines from each token, but store in the tuple for each token whether it originally ended in a new-

```

...
(11765, 'calculated', False, []), (11766, 'using'      False, []),
(11767, 'the',          False, []), (11768, 'Relative',  False, []),
(11769, 'Value,',      False, []), (11770, 'at',         False, []),
(11771, 'Risk',        False, []), (11772, 'Approach',  False, []),
(11773, '.',           True,  []),  (11774, 'Portfolio', False, []),
(11775, 'management', False, []), (11776, 'and',         False, []),
(11777, 'investment',  False, []), (11778, 'advising',   True,  []),
(11779, 'UBS',         False, []), (11780, 'Third',     False, []),
(11781, 'Party',       False, []), (11782, 'Management', False, []),
(11783, 'Company',     False, []), (11784, 'S',         False, []),
(11785, '.',           True,  []),  (11786, 'A',         False, []),
(11787, '.',           True,  []),  (11788, '.',         False, []),
...

```

Figure 1: Enriched Document Tokens before Automatic Annotation

line. Note that all period (.) tokens in the data have a trailing newline, apparently added at tokenization time, while other newlines are merely for layout purposes (e.g. headlines, cf. token 11778 in Figure 1). Finally, each tuple contains an (initially empty) list to which annotation labels are added. Figure 1 contains a short excerpt from the Dev data set.

Then, we perform an automatic, pattern-based annotation of the input document, which assigns descriptive labels (e.g. URL, ABBREVIATION, ENUMERATION, and SECTION) to sequences of tokens. Apart from providing shallow hints regarding the document structure (e.g. start / end of an ENUMERATION, which normally occur in groups), these labels also effectively disambiguate potential *ES* markers (mainly periods (.) by binding them to the larger units. Since neither an annotation scheme nor an operationalizable definition of *BS* and *ES* for the domain of the Fin-SBD documents were available, rules were created inductively by analyzing the annotated documents (based on an HTML-based visualization). Finally, we perform the actual *ES* and *BS* detection, which employs hand-crafted, heuristic rules partly operating on the plain tokens, and partly on the token annotations. While the heuristic rule application happens *after* the pattern-based annotation, we describe it first (Section 5.1), and the pattern-based annotation afterwards (Section 5.2).

5.1 ES and BS Detection

ES and *BS* detection work by going through the list of 4-tuples and checking each tuple against a short list of rules. These rules take into account the string and pertaining annotations of both the current token T_i and its immediately preceding and following tokens T_{i-n} and T_{i+n} . *ES* detection is done first, because it provides some information that is used by *BS* detection later. The full list of heuristic rules for *ES* detection in their actual order is given in Figure 2. At most *one* rule is applied to every token T_i . We use the expression $\text{STRING}(T_i)$ to represent the actual token, $\text{ANNO}(T_i)$ to represent the annotations assigned to T_i , and $\text{NEWLINE}(T_i)$ to represent whether the token originally ended in a newline. Many of the rules in Figure 2 are more or less self-explanatory. Rules 1 and 2 directly handle potential *ES*-signalling tokens, with the additional condition in rule 1 ensuring that a previously disambiguated period character (.) is prevented from causing a sentence break. Rule 3 exploits an observed structural property of the annotation, i.e. that items in enumerations (like ‘(a) ...’, ‘(b) ...’, ‘(c) ...’) are treated as sentences. Similarly, rule 4 exploits the fact that token se-


```

Given a token  $T_i$ , label it as ES if
1.  $STRING(T_i) = '-'$  and
    $ANNO(T_i) = []$ ;
2.  $STRING(T_i) = '?'$  or  $!'$ ;
3.  $ANNO(T_{i+1}) = 'B-ENUM'$ ;
4.  $STRING(T_i) = ':'$  or  $'and'$  or  $'or'$  and
    $STRING(T_{i+1}) = '-'$ ;
5.  $STRING(T_i) = ':'$  or  $'$  or  $'$  or  $'and'$  or  $'or'$  and
    $ANNO(T_{i+1}) = 'B-ENUM'$  and
    $ANNO(T_{i-1}) != 'E-ENUM'$ ;

```

Figure 2: Heuristic ES Detection Rules

quences starting with a dash (-) are also treated as sentences (cf. also rule 4 in Figure 3), and that the tokens ',', 'and', and 'or' also mark sentence breaks if they immediately precede such a dash. Rule 5, finally, handles sequences of enumerations. For *BS* detection, the rules are given in Figure 3. Note that tokens T_i for which $ANNO(T_i) = 'B-HEADLINE'$ or 'B-BULLET' are regarded as non-*BS*, and are not submitted to the rules. The same is true for tokens T_i for which $STRING(T_{i-1}) = '-'$ and $NEWLINE(T_{i-2}) = True$.

```

Given a token  $T_i$ , label it as BS if
1.  $ANNO(T_i) = 'B-ENUM'$ ;
2.  $ANNO(T_{i-1}) = 'E-HEADLINE'$ ;
3.  $ANNO(T_{i-1}) = 'E-BULLET'$  and
    $ANNO(T_i) = []$ ;
4.  $STRING(T_i) = '-'$  and
    $ANNO(T_{i-1}) = 'ES'$ ;
5.  $STRING(T_i)$  starts_with_uppercase and
    $ANNO(T_{i-1}) = 'ES'$ ;

```

Figure 3: Heuristic BS Detection Rules

The rules in Figure 3 are less complex than those in Figure 2. At least in part, this is due to the fact that they make use of the results of previous rules (cf. below). The first two rules in Figure 3 are very straightforward. One striking difference between the *BS* and the *ES* detection rules is that the former makes use of the output of the latter: Rule 4 is the complement to rule 4 in Figure 2, which assigns the *BS* label to dash characters (-) which immediately follow a previously assigned *ES* label. Similarly, rule 5 is a kind of default which assigns the *BS* label to all tokens with an uppercase initial character directly following a previously assigned *ES* label.

5.2 Pattern-based Automatic Annotation

In this step, the document is matched against a group of simple to averagely complex patterns. As mentioned earlier, this step happens *before* rule application, because the patterns are used to enrich the input for the rules. Some patterns operate strictly locally, while others depend on earlier patterns. The patterns, their sequence of application, and their respective *cumulative* effects on rule application for Train, Dev, and Test can be found in Table 3.

Note that the result precision was set to five decimal places because, as will become apparent, the effects that some patterns have on *BS* and *ES* detection are rather subtle. Row 0 contains the results when the rules described in Section 5.1 are applied without any previous annotation. Performance

differences between Train, Dev, and Test on this level are indicative of inherent differences between these data sets. Actually, we can observe the following: i) On Train, F score for the *BS* task is higher than for the *ES* task, while on Dev and Test, F score for the *ES* task is higher. ii) Looking only at the *BS* task, P and R are *roughly* on a par for all three data sets, while for the *ES* task, R is generally much higher than P, and F scores for Train and Test are very similar (.67758 and .66194), while F score for Dev is much higher (.74945). iii) Looking only at the *ES* task, F score for Dev and Test are reasonably similar (.82627 and .80643), while F score for Train is much lower (.62658).

The URLs and DATES patterns (rows 1 and 2) detect simple expressions like 'www . ubs . com' and '12 . 10 . 2011', respectively. These are mainly targeted at period (.) disambiguation, which should be visible in improvement of P for *ES* detection. And in fact, *BS* performance on neither Train, Dev, nor Test is affected by these patterns, and *ES* R is also constant. We can see the expected improvements in P for *ES* detection, however, these are extremely small only.

The ABBREVS_LU and ABBREVS_PT patterns (rows 3 and 4) detect the abbreviations including period (.) characters. The first pattern does a simple look-up in a predefined list of abbreviations, while the second detects the abbreviations by matching sequences of single upper case letters and period characters. These two patterns are also mainly targeted at period disambiguation, which is indeed visible in considerable jumps in *ES* detection P (.49848 to .56340 for Train, .75073 to .89815 for Dev, and .73839 to .77070 for Test). R is consistently drops a little, but overall F score for *ES* detection increases. However, we also see a positive effect of these two patterns on P of the *BS* detection task, because with more correct *ES* being detected, the coverage of the *BS* detection rules (e.g. rule 5 in Figure 3) also improves.

The SECTION pattern (row 5) is similar to the ABBREVS_PT pattern, but detects sequences of numbers and period characters. This is another pattern targeting period disambiguation which should have an impact on P for *ES* task. However, this is the case for Train only, where P and consequently F score is considerable improved from .67546 to .73884, while there is hardly any impact on Dev and Test.

The AMOUNTS pattern (row 6) is a variant of the SECTION pattern which also handles leading and trailing zeros, to detect expressions such as '0 . 025'. This pattern has the overall expected positive effect on P (and F score) for the *ES* detection task, but again, we observe huge differences in improvement between Train on the one hand and Dev and Test on the other. For Train, application of this pattern boosts P of *ES* from .66324 to .83619 with only a marginal drop in R, resulting in an improvement in F score from .73884 to .83473. For Dev and Test, the final improvement in F score is only about .02.

ENUMS and ENUMS-MERGE (rows 7a and 7b) is a two-step pattern which does not address period disambiguation, but the detection of expressions like '(a', '(a)', 'I b)' etc., which the Fin-SBD annotation treats as sentences. Accordingly, this pattern has a more complex effect, as it mainly addresses R for both *BS* and *ES*, while accepting some drops in P. In general, however, application of this pattern(s) results

Pattern	Train						Dev						Test					
	P	BS R	F	P	ES R	F	P	BS R	F	P	ES R	F	P	BS R	F	P	ES R	F
0 -	.68976	.66583	.67758	.49488	.85382	.62658	.76330	.73627	.74954	.74810	.92269	.82627	.65909	.66482	.66194	.73550	.89249	.80643
1 URLS	.68976	.66583	.67758	.49843	.85382	.62943	.76330	.73627	.74954	.74897	.92269	.82680	.65909	.66482	.66194	.73839	.89249	.80816
2 DATES	.68976	.66583	.67758	.49848	.85382	.62947	.76330	.73627	.74954	.75073	.92269	.82788	.65909	.66482	.66194	.73839	.89249	.80816
3 ABBREVS_LU	.72387	.66059	.69079	.54893	.84585	.66579	.76646	.72327	.74424	.80608	.91908	.85888	.65986	.66403	.66194	.75300	.89170	.81650
4 ABBREVS_PT	.74823	.65912	.70085	.56340	.84317	.67546	.83207	.71604	.76971	.89815	.91113	.90459	.66933	.66245	.66587	.77070	.89012	.82612
5 SECTIONS	.76025	.65075	.70125	.66324	.83390	.73884	.83207	.71604	.76971	.90394	.91113	.90752	.66933	.66245	.66587	.77495	.89012	.82855
6 AMOUNTS	.76025	.65075	.70125	.83619	.83327	.83473	.83404	.71532	.77013	.93824	.91113	.92449	.66933	.66245	.66587	.80300	.88933	.84396
7a ENUMS	.71812	.73673	.72730	.79389	.89925	.84329	.78777	.79118	.78947	.89708	.97616	.93495	.62608	.74387	.67991	.76662	.94783	.84765
7b ENUMS-MERGE	.73191	.72961	.73076	.81434	.89893	.85455	.81111	.79118	.80102	.92156	.97616	.94807	.64408	.74387	.69039	.78830	.94783	.86073
8 HEADLINES	.80346	.85413	.82802	.81327	.88443	.84736	.89393	.88295	.88840	.92156	.97616	.94807	.74251	.86166	.79766	.79024	.94704	.86156
9 BULLETS	.80199	.84245	.82172	.81126	.86939	.83932	.90973	.93208	.92077	.92324	.97327	.94759	.74251	.86166	.79766	.79024	.94704	.86156

Table 3: Results of BS and ES Detection Rules with Cumulative Effect of Different Annotation Patterns

in an increase in F for both *BS* and *ES* for all data sets.

The HEADLINES pattern (row 8) uses, among other things, the NEWLINE feature extracted from the raw data (cf. Figure 1 above) to distinguish actual sentences from sentence-like headlines, which are not annotated as sentences in the Fin-SBD data. Headline information is explicitly used for *BS* detection (cf. rule 2 in Figure 3). Accordingly, this pattern mainly affects the *BS* results, while *ES* results are mostly unaffected. It greatly improves both P and R for *BS* on all three data sets, with increases of up to .1 in F: F for Train increases from .73076 to .82802, for Dev from .80102 to .88840, and for Test from .69039 to .79776.

The final pattern is BULLETS (row 9), which looks for itemized text. The effect of this pattern, however, is mixed, with some small increases in R for some data sets, and some decreases in others, but no effect at all on the Test set.

Of the two final results that we submitted, the first one (HITS-SBD1) is the final result for Test in Table 3, which was rounded and averaged by the organizers to an F of .83.

5.3 Optional PDF Re-Processing

The patterns and rules described and analyzed above represent the core of our rule-based system. We created one alternative result (submitted as HITS-SBD2) in which we addressed an apparent problem with the original tokenization, which we expected to improve the results considerably. During the data set inspection, we often observed cases where tokens in the original data set were incorrectly merged, like in the following examples:

```
(31288, 'the', [], False),
(31289, 'Prospectus', [], False),
(31290, '.', [], True),
(31291, 'Investor', [], False),
(31292, 'profile', [], True),
(31293, 'The', [], False)
...
(38074, 'duties', [], False),
(38075, 'or', [], False),
(38076, 'other', [], False),
(38077, 'chargesD', [], False),
(38078, '=', [], True),
(38079, 'net', [], False)
```

We applied Apache tika¹ to the provided PDF files and created an improved tokenization. We automatically aligned it with the original tokenization and detected cases of incorrect token merges. These were then split, where care was taken to retain the original token indices. As a result, the improved tokenization looked like the following:

```
(31288, 'the', [], False),
(31289, 'Prospectus', [], False),
(31290, '.', [], True),
(31291, 'Investor', [], False),
(31291, 'profile', [], True),
(31292, 'The', [], False)
...
(38074, 'duties', [], False),
(38075, 'or', [], False),
(38076, 'other', [], False),
(38077, 'charges', [], False),
(38077, 'D', [], False),
(38078, '=', [], True),
(38079, 'net', [], False)
```

While we were expecting this improved pre-processing to have a huge effect on our results, the actual improvements were minimal: On Test, we obtained an F of .80162 for *BS* and .86280 for *ES*, respectively.

6 Summary and Conclusion

We presented two competing systems for SBD that were developed for the Fin-SBD shared task 2019. The ML-based system was based on a simple and elegant approach which was inspired by the prior work on SBD in more classical, less noisy genres, where more locally oriented features (like the context windows applied in this work) are known to work better. However, in the domain of the Fin-SBD shared task, where the notion of *sentence* is less well-defined, this approach failed to reach an acceptable result. The other approach was based on a combination of simple, surface-based patterns, and heuristic rules for solving the task in an unsupervised manner. Both patterns and rules were created manually on the basis of the introspection of the Train and Dev data sets. While this approach allowed for the creation of some high-precision rules, including ones that are strongly tailored towards the sometimes idiosyncratic annotations in the original data set, it failed to produce a complete and sufficiently robust solution. However, the results, while more in the bottom range, are still acceptable, especially given the fact that the results from all competing parties are rather close together.

Acknowledgements

This work has been supported by the Klaus Tschira Foundation, Heidelberg, Germany. In addition, the first author has been supported by the German Academic Exchange Service (DAAD), Germany, and Higher Education Commission (HEC), Pakistan.

¹<https://tika.apache.org/>

References

- [Agarwal *et al.*, 2005] Neha Agarwal, Kelley Herndon Ford, and Max Shneider. Sentence boundary detection using a maxEnt classifier, 2005.
- [Ait Azzi *et al.*, 2019] Abderrahim Ait Azzi, Houda Bouamor, and Sira Ferradans. The FinSBD-2019 Shared Task: Sentence boundary detection in PDF Noisy text in the Financial Domain. In *The First Workshop on Financial Technology and Natural Language Processing (FinNLP 2019)*, Macao, China, 2019.
- [Akita *et al.*, 2006] Yuya Akita, Masahiro Saikou, Hiroaki Nanjo, and Tatsuya Kawahara. Sentence boundary detection of spontaneous Japanese using statistical language model and support vector machines. In *Ninth International Conference on Spoken Language Processing (IC-SLP)*, pages 1033–1036, Pittsburgh, Pennsylvania, 2006. International Speech Communication Association.
- [Gillick, 2009] Dan Gillick. Sentence Boundary Detection and the Problem with the U.S. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244, Boulder, Colorado, 2009. Association for Computational Linguistics.
- [Joachims, 1997] Thorsten Joachims. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*, pages 143–151, Nashville, TN, USA, 1997. Morgan Kaufmann Publishers Inc.
- [Jonathon *et al.*, 2012] Read Jonathon, Dridan Rebecca, Oepen Stephan, and Jørgen Solberg Lars. Sentence Boundary Detection: A Long Solved Problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India, 2012. The COLING 2012 Organizing Committee.
- [Kiss and Strunk, 2006] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.
- [Kreuzthaler and Schulz, 2015] Markus Kreuzthaler and Stefan Schulz. Detection of sentence boundaries and abbreviations in clinical narratives. *BMC Medical Informatics and Decision Making*, 15(2):S2–S4, 2015.
- [Orosz *et al.*, 2013] György Orosz, Attila Novák, and Gábor Prószéky. Hybrid text segmentation for Hungarian clinical records. In *12th Mexican International Conference on Artificial Intelligence*, pages 306–317, Mexico City, Mexico, 2013. Springer.
- [Polanyi *et al.*, 2004] Livia Polanyi, Chris Culy, Martin Van Den Berg, Gian Lorenzo Thione, and David Ahn. A rule based approach to discourse parsing. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*, pages 108–117, Cambridge, Massachusetts, USA, 2004. Association for Computational Linguistics.
- [Reynar and Ratnaparkhi, 1997] Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *ANLC '97: Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19, Washington, DC, 1997. Association for Computational Linguistics.
- [Rudrapal *et al.*, 2015] Dwijen Rudrapal, Anupam Jamatia, Kunal Chakma, Amitava Das, and Björn Gambäck. Sentence Boundary Detection for Social Media Text. In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 254–260, Trivandrum, India, 2015. NLP Association of India.
- [Wiecheteck *et al.*, 2019] Linda Wiecheteck, Sjur Nørstebø Moshagen, and Thomas Omma. Is this the end? Two-step tokenization of sentence boundaries. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 141–153, Tartu, Estonia, 2019. Association for Computational Linguistics.

PolyU_CBS-CFA at the FinSBD task: Sentence Boundary Detection of Financial Data with Domain Knowledge Enhancement and Bilingual Training

Mingyu Wan^{1*}, Rong Xiang², Emmanuele Chersoni¹, Natalia Klyueva¹, Kathleen Ahrens³, Bin Miao⁴, David Broadstock⁴, Jian Kang⁴, Amos Yung³ and Chu-Ren Huang¹

¹Department of Chinese and Bilingual Studies, The Hong Kong Polytechnic University

²Department of Computing, The Hong Kong Polytechnic University

³Department of English, The Hong Kong Polytechnic University

⁴School of Accounting and Finance, The Hong Kong Polytechnic University

{mingyu.wan, kathleen.ahrens, bin.miao, david.broadstock, jian.kang, amos.yung, churen.huang}@polyu.edu.hk, {xiangrong0302, emmanuelechersoni, natalka.kljueva}@gmail.com

Abstract

Sentence Boundary Detection is a basic requirement in Natural Language Processing and remains a challenge to language processing for specific purposes especially with noisy source documents. In this paper, we deal with the processing of scanned financial prospectuses with a feature-oriented and knowledge-enriched approach. Feature engineering and knowledge enrichment are conducted with the participation of domain experts and for the detection of sentence boundaries in both English and French. Two versions of the detection system are implemented with a Random Forest Classifier and a Neural Network. We engineer a fused feature set of punctuation, digital number, capitalization, acronym, letter and POS tag for model fitting. For knowledge enhancement, we implement a rule-based validation by extracting a keyword dictionary from the out-of-vocabulary sequences in FinSBD's datasets. Bilingual training on both English and French training sets are conducted to ensure the multilingual robustness of the system and to extend the relatively small training data. Without using any extra data, our system achieves fair results on both tracks in the shared task. Our results (English¹: F1-Mean = 0.835; French: F1-Mean = 0.86) as well as a post-task quick improvement with self-adaptive knowledge enhancement based on testing data demonstrate the effectiveness and robustness of bilingual training with multi-feature mining and knowledge enhancement for domain-specific SBD task.

*Contact Author

¹This is the adapted result as illustrated in Section 5.

1 Introduction

Sentence Boundary Detection (SBD), which aims at detecting/disambiguating sentence boundaries of texts, is a fundamental step in many Natural Language Processing (NLP) applications. It should be carried out before other critical components of NLP, *e.g.* part-of-speech (POS) tagging, syntactic-semantic-discourse parsing, information extraction or machine translation. Existing SBD approaches have shown promising results for languages that have dependable orthographic conventions to mark beginning and ending of sentences, such as in English and many European languages. Relevant recent work include (*e.g.* [Riley, 1989; Reynar and Adwait, 1997; Mikheev, 2002; Palmer and Hearst, 1997; Read, 2012]). However, previous work in SBD mainly dealt with well-formed and clean data such as articles from the Brown corpus [Hearst, 1994] or Wall Street Journal [Palmer and Hearst, 1997].

SBD remains challenging in two scenarios. The first involves documents encoded in non-text formats, such as Adobe PDF format, or other image formats. They provide the exact layout of a human readable document on a wide range of machines. However, texts converted from PDF documents by OCR software are usually noisy and potentially with the loss of substantial formatting features. This chaos leads to difficulties in SBD, and so far has been under-researched. The second involves languages whose orthography does not mark sentence boundary conventionally. For instance, [Huang and Chen, 2017] shows that using the period punctuation will lead to significant divergence from sentence boundaries. What they proposed are followed by [Hou *et al.*, 2019] in their Menzerath-Altamann based power relations between a clause and its constituent words (instead of between sentences and words). In this current paper, we deal with the first challenge.

There are a number of issues to be addressed when applying SBD to financial documents. Unlike passages of

formal texts, financial documents are often heavily populated with rich tables of data—sometimes stretching over multiple pages—and figures, titles, dates and keywords of various types. The presence of such non-textual information is admittedly not unique to financial documents, but it should be noted that many financial documents also do not come in clean/easily machine readable structures. Detecting sentence boundaries on the basis of periods/stops may also be less straightforward, for example the presence of company tickers in a document may introduce some difficulties in cleanly identifying sentence boundaries, especially if appended with exchange, for example the full ticker for China Light and Power (CLP) company listed in Hong Kong can be written as ‘0002.hk’. Additionally, sentences may contain various financial terms/acronyms, including company name abbreviations, that may impact the syntactic structure and hence generate sentence confusion.

As such financial documents present a range of challenges that result in the need to use a hybrid of language processing tools in combination with knowledge enrichment specific to the financial domain. With such endeavors, we can expect chances of achieving SBD with reasonable levels of accuracy.

In the following sections, we will review some related work in Section 2, describe the features and methodology in Section 3, show and discuss the results in Section 4 and finally conclude this work in Section 5.

2 Related Work

Sentence Boundary Detection is a fundamental issue in Natural Language Processing, which can be viewed as a classification issue. Current studies normally tackle the problem as the identification of the truthful sentence ending markers among the ambiguous ones. The history of SBD development witnessed machine learning as the earliest attempts (*e.g.* [Riley, 1989; Reynar and Adwait, 1997; Palmer and Hearst, 1997]), with rule-based systems coming afterwards (*e.g.* [Mikheev, 2002; Mikheev, 2000]). There has been some work occasionally using unsupervised techniques (*e.g.* [Kiss and Strunk, 2006]).

Early attempts have already shown promising results of SBD, but all with well-formed data. For example, since Riley [1989]’s very first work of SBD, a 99.8% accuracy was reported by investigating only a single punctuation mark, *i.e.* period, with the use of Decision Tree classifier trained on 25 million words of newswire texts. Hearst [1994] achieved a 1.5% error rate by using Feedforward Neural Network of POS features trained on the Brown corpus. Later on, Palmer and Hearst [1997] developed SATZ, a system that used features of contextual POS distribution and words-as-vectors of the target punctuations *via* NN and DT with training on the 30 million WSJ corpus. Their work hit the record of the state-of-the-arts of SBD with error rates of 1.1% for NN and 1% for DT. Synchronically, Reynar and Adwait [1997] adopted supervised Maximum Entropy learning with two

sets of features: in-domain financial uses, *e.g.* honorifics (Mr., Dr., etc.) and corporate designations (Corp.); domain-independent abbreviations, as well as ‘!’, ‘.’ or ‘?’ as potential boundaries. This work, however, was slightly inferior in performance with accuracies of 98.8% and 97.9% respectively for the domain-dependent system and accuracies of 98.0% and 97.5% respectively on the portable system.

Modern machine learning techniques provide us with a series of statistical models focusing on data patterns, nonlinear features and forecast accuracy. With the breakthrough of computing technology, the nonlinear methods became feasible in 1980s, as represented by Breiman *et al.* [1984]’s work with tree-based and regression models. Since then, an increasing number of tree-based models, both supervised and unsupervised, were developed and promptly emerged, such as Random Forests, Boosting Trees, *etc.* Prior to traditional classifiers, Neural Network methods were introduced to SBD by McCulloch and Pitts [1943]. From 1980s, the Neural Network, incorporating the Bayesian Neural Network, was resurged by the upgrades of computing technology as well as the appearance of back-propagation algorithms. Unlike tree-based methods, NN methods present smooth functions of parameters, which facilitate the development of Bayesian inference.

Complementary to machine learning, Mikheev [2000; 2002] employed rule-based systems for SBD, and reported error rates of 0.31% and 0.20% with training on WSJ and the Brown corpus respectively. Recently, Read *et al.* [2012] and Griffis *et al.* [2016] both adopted several state-of-the-art NLP toolkits for SBD with mixed datasets of varied formality and specificity. Both works showed that the existing toolkits for SBD in specific domains are worse without resistance to domain-transfer or formality change.

To approach the above-mentioned problem of SBD in recent decade, we need to enforce renewed efforts of further shaping the NLP tools as well as addressing to domain-specific and informality issues, with aim of refreshing a new record in the SBD history.

In this paper, we propose a feature-oriented and knowledge-enriched approach to detecting the beginnings and endings of financial data in both English and French by using a Random Forest Classifier (RFC) and a Neural Network (NN). In addition to feature engineering, model fitting and parameter tuning, we conduct post-classification knowledge enhancement with a rule-based keyword validation on the predictions by automatically identifying and extracting the out-of-boundary word sequences from the FinSBD datasets [Ait Azzi *et al.*,]. In addition, the main body of noise in the datasets provides us with a useful resource as a by-product of the task for rectifying the ambiguous boundaries.

3 Features and Methodology

In this section, we describe the features and methodology of this work with a pipeline of feature engineering,

classification and aspect knowledge enrichment (post-classification validation).

3.1 Feature Engineering

Feature engineering plays an important role in machine-learning, involving the selection of a subset/fused set of informative and discriminative features with dual purposes of dimension reduction and classification leverage [Garla and Brandt, 2012]. In general classification tasks, features typically include bags of characters, words, n-grams and/or concepts in a text corpus, which, however, causes high dimensionality of feature space in lowering classification efficiency.

Feature selection is necessary when feature space is overloaded or in redundancy. Algorithms of term frequency, chi-square, information gain, mutual information or relevance score are usually adopted in automatic feature selection (*e.g.* [Lee and Lee, 2006; Chen *et al.*, 2009]); domain knowledge is also helpful to guide the feature engineering process. In this sentence segmentation task, we utilize a semi-automatic selection method that both considers high frequency words and keyword knowledge, as shown in Sections 3.1 and 3.3 below.

By a close observation and comparison of the scanned documents with the gold boundaries in the datasets [Ait Azzi *et al.*,], we introspect the key sections of erroneous predictions both manually and statistically. This leads to the inclusion of the following sets of salient features for fitting the classifiers.

- **Two sets of punctuation:** Punctuation serves as important cues for SBD and has been proved as the most useful feature in SBD. In addition to using period as the baseline feature set, we also include a set of special punctuation-symbols that are prevalent in financial data, such as the dollar signs, math operators and copyright symbols, as listed below:

- PUNC_SET1 = [':']
- PUNC_SET2 = ['?', '!', ',', '%', '-', '/', '"', '\', ')', '(', '*', '□', '<', '>', '≤', '•', 'e', '\$', 'ℒ', "“”, '©', '®']

By adding the second punctuation set in the attribute table, we got 1% F1 improvement for the validation sets of both languages.

- **Initially capitalized words:** As suggested in the samples of gold boundaries, most BEGINS are marked with initially capitalized words (*e.g.* “Distribution-BEGIN”) or the ENDS are largely preceding such words (*e.g.* “.Enter-END The-BEGIN sales”). Although it introduces some confusing information for the classifiers, such as the keywords in titles, tables, figures, etc., the inclusion of this type of feature on average improves 2% F1 score of validation for both languages. In order to associate such feature with both BEGINS and ENDS, we use a feature array of three dimension in the pre-, current, post- positions to maximally represent its discriminativeness in predicting boundaries.

- **Acronyms or Abbreviations:** Acronyms or abbreviations are also salient features for marking boundaries as indicated in both the existing literature and the FinSBD datasets. For example, “UBS_BEGIN” “co” or “kiid” show that acronyms tend to co-occur (all capitalized words) or not occur (all lower cased words) with boundaries. As such, we construct a three-dimension attribute array of storing the Boolean value of all-word-capitalization in the pre-, current, post- positions. This feature set also improves around 1% F1 in the validation set for both languages.

- **Digital numbers:** Digital numbers is a common property of financial data which causes confusion for disambiguating *e.g.* decimal points from endings, as in “10.3”. To identify both the left and right context of the target period, we also construct a three-dimension attribute array for representing such cases. This feature set helps improve 2% F1 for the French validation set and 1% F1 for the English validation set.

- **Letters or Roman numbers:** As another salient feature in financial data, letters (*e.g.* A-Z, a-z alphabetical letters) or Romance numbers (I, II, . . . , XII) are highly suggestive of non-boundary tokens. Therefore it also serves as a useful feature for excluding the wrong boundaries. A tri-gram feature array is also constructed to represent such information in the pre-, current, post- positions, which helps around 1% F1 improvement of both validation sets.

- **POS tags:** Despite the fact that sentence segmentation occurs prior to part-of-speech tagging in NLP processing, the pos information of individual tokens can, in turn, indicate the phrasal structure of a word sequence which may provide useful cues to the identification of verbal sentences or alternatively, the non-clausal noun phrases (keywords). By including the three-dimension POS feature (the UPOS tag set by UDPipe²) in our experiment, our system is further optimized with 3% F1 increase for both validation sets.

- **Enter ('\n'):** Enter ('\n') seems a universal feature for any type of document. But after a close look into the converted pdf documents in the FinSBD datasets, we found that Enters ('\n') is strongly associated with the conversion errors caused by the pdf scanning. With the inclusion of such feature in a three-dimension array, we further improve the system with 1% F1 for both validation sets.

For maximizing the discriminative power of the above features, we construct a fused feature set of 24 dimensions to fit the machine learning models and get an optimized performance (English: F1-Mean = 0.87; French: F1-Mean = 0.85) in the validation sets.

²<http://ufal.mff.cuni.cz/udpipe>

3.2 Classification Models

Ensemble Learning of Random Forest

Random Forest Classifier (RFC) is a tree-based ensemble classifier. It combines the decision of multiple Decision Tree (DT) classifiers where each classifier is generated using a random vector independently sampled from the input vector, and each tree casts a unit vote for the most popular class to classify an input vector [Breiman, 1999].

The ensemble RFC is generally more accurate than all the individual classifiers as it makes use of many naive classifiers that randomly use a subset of the vector, thus it is more robust to overfitting in comparison to traditional decision trees. As such, it is our first choice of classifier in this study. The RFC classifier is imported from the sklearn package³ where a random forest is taken as a meta estimator (n_estimators is 10 by default) that fits a number of decision tree classifiers on various subsamples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

The design of a decision tree required appropriate attribute selection measure and a pruning method. In our experiment, we use randomly selected features at each node to grow a tree with an optimized setting of min_samples_split as 8, max_features as “log2” and random_state as 10. In addition, we set oob_score true to use out-of-bag samples to estimate the generalization accuracy.

The above setting of parameters of RFC work out the best performance in our estimation on the validation sets.

Neural Network

Resembling the biological neural networks, artificial Neural Networks (NN) approaches were proposed and led to great improvements in a number of NLP tasks. An artificial NN is usually composed of many simple processors (neurons) that are interconnected, operate in several layers and learn from input of examples. Considering the similar characteristics of financial data, we implemented a NN-based approach as a complementary work to the RFC model.

In the validation, we trained the Multi-layer Neural Network using Tensorflow⁴ following several runs of parameter optimization. The optimization was done with bilingual training on both English and French training sets and testing on the English validation set. The optimal setup for training the model was a network of one input layer (density 300) and 1 hidden layer (density 100) with the relu activation function, and one output layer (3 categories) with the softmax activation function. As a loss function we used categorical crossentropy, and Adam as an optimizer. The batch size was set to 32, and the number of epochs was 5.

The input units were the feature vectors as described in Section 3.1, but certain features were excluded for the

³<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

⁴https://www.tensorflow.org/api_docs/python/tf/nn

Neural Network run of the French trial for reasons of optimization.

3.3 Domain Knowledge Enhancement

Knowledge enrichment has been proven to be a useful guidance for the post-processing of confused classification by statistical models [Ghosh and NAG, 2002]. In this task, OCR conversion of financial documents introduced large chunks of out-liars, such as the titles, dates, tables, figures, etc., which fail to fall into the traditional category of sentence boundary. These non-textual sections, as finely segmented with the “Enter” marks, cause erroneous predictions of boundaries. To solve this problem, we implemented a post-processing procedure to correct false positive predictions, as realized with the following two algorithms.

Keyword Extraction

Following the above-mentioned principle, we constructed a keyword dictionary with Algorithm 1. A broader definition of keyword is adopted here, including out-of-boundary words, symbols and phrases. We utilize the well segmented training and validation sets [Ait Azzi *et al.*,] for resource construction. Intuitively, if any word sequence locates between an “END” and the next “BEGIN”, we regard it as a potential out-lier and construct keywords with further segmentation marked by “Enter”. The final keyword dictionary with respect to both languages is then constructed, containing elements of key-value (keyword-frequency) pairs.

Algorithm 1 Keyword Extraction

Input: dataset

Output: keyword_dict

```
1: for i in len(dataset) do
2:   curword = dataset[i].
3:   nxtword = dataset[i+1].
4:   if “END” in curword then
5:     if “BEGIN” in nxtword then
6:       continue
7:     else
8:       add keyword to keyword_dict.
9:       update frequency in keyword_dict.
10:    end if
11:  end if
12: end for
13: refine keyword_dict with length threshold.
14: refine keyword_dict with frequency threshold.
15: return keyword_dict
```

To further control the quality of extracted keywords, we introduced *length threshold* and *frequency threshold* to filter those patterns that are too short or rarely occurring. As a result, a keyword dictionary of 16,501 keyword-frequency pairs is generated for the rule-based validation, as well as providing a financial resource for potential use in future IR inquiries.

Rule-based Validation

With the keyword dictionary generated, we used a rule-based approach for correcting the potentially wrong boundaries that are not in the keyword list, as illustrated in Algorithm 2.

Algorithm 2 Rule-based validation

Input: dataset, raw_pred, keyword_dict

Output: updated_prediction

```
1: for keyword in keyword_dict.key do
2:   for i in len(dataset)-len(keyword) do
3:     if word sequence match keyword then
4:       update raw_pred with NO_BOUNDARY
5:     end if
6:   end for
7: end for
8: return raw_pred
```

As shown in Algorithm 2, each keyword in the dictionary is used as a rule. Every word sequence that matches a keyword in the dictionary shall be forced to the “NO_BOUNDARY” class. This process will be executed iteratively until all the predictions are validated. As to be shown in the following section, the experimental results on the validation sets and the final results on the test sets of both languages have consistently verified the usefulness of knowledge enhancement in domain-specific classifications.

4 Results

In this section, we show our classification results with the following four aspects of comparisons.

4.1 Classifiers

This section focuses on the comparison of the classification performance of the two classifiers, *i.e.* RFC and NN, with the same experimental setting. A fused feature set is used and bilingual training is conducted. The classification results on both the validation (Dev) set and the test set of the two languages are shown in Figure 1 below:

As it is easy to see in Figure 1, RFC shows superior performance with 1% or 4% F1 gain compared to NN for both the validation tasks (Dev_en and Dev_fr) and the testing of the French track (Test_fr). This is highly suggestive that the ensemble random forest is more fitted to the selected feature set in this work, while NN seems to demonstrate no advantage of winning traditional classifiers despite having the same salient feature set in this task.

However, what is contradictory to our estimation is: NN outperforms RFC with 3.5% F1 discrepancy in the English track (Test_en), whereas the results of our validation on the English development set is opposite (RFC: 0.875 vs. NN: 0.86). Another obvious observation is: both classifiers’ performance on the English test set

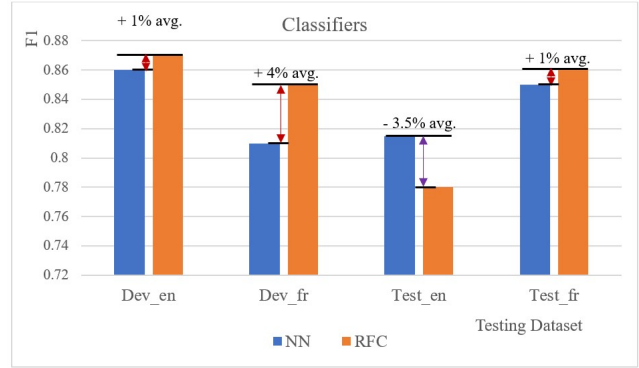


Figure 1: Classification Results of NN vs. RFC

drops significantly with a slip of 4-10% F1. By reviewing the released test set with gold labels, we found that a large number of new acronyms and code series are introduced. This causes a systematic deterioration of both classifiers, but NN presents a higher robustness to the feature surprise.

The unexpected performance of RFC in the English test set can be attributed to an over-fitting problem and hence draws our attention to look for a semi-supervised or un-supervised mechanism in complementing the feature mismatch between the validation set and the test set, which shall lead to a more stable result in similar tasks.

4.2 Bilingual Training

This section focuses on the comparison of the classification performance of bilingual vs. monolingual training with the same experimental setting. A fused feature set and the RFC classifier is adopted. As what we submitted for the contest are all based on bilingual training, the current comparison is based on the validation (Dev) set only. The classification results on the validation set of the two languages are shown in Figure 2 below:

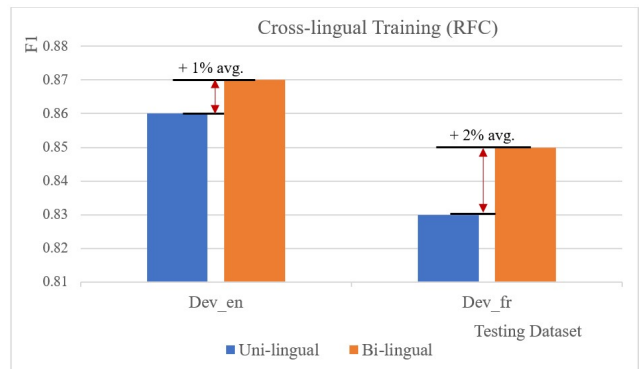


Figure 2: Classification Results of bi- vs. mono-lingual training

As shown in Figure 2, bilingual training brings a consistent benefit to the classification performance with 1%

F1 improvement for English and 2% F1 improvement for French. This impact can be counted as huge for any kinds of competition. By referring to this set of validation results, we finally conduct bilingual training for all the runs of submission.

4.3 Features

This section focuses on the comparison of the feature discriminativeness in the SBD classification task by adding the individual feature set separately in each implementation. The basic experimental setting is the same, including using the RFC classifier and bilingual training. As the submissions are all based on the full feature set, the current comparison is implemented on the validation (Dev) set only. The classification results on the validation set of English are shown in Table 1 below:

Features\F1	BS ^a	ES ^b	Mean	$\delta(\%)^c$
Punc1	0.70	0.82	0.76	baseline
+Punc2	0.71	0.83	0.77	1 \uparrow
+Cap	0.72	0.86	0.79	2 \uparrow
+Acro	0.73	0.87	0.80	1 \uparrow
+Dig	0.73	0.89	0.81	1 \uparrow
+Lett	0.74	0.90	0.82	1 \uparrow
+POS	0.78	0.92	0.85	3 \uparrow
+Enter/All	0.80	0.92	0.86	1 \uparrow

^a Beginning boundaries

^b Ending boundaries

^c F1 improvement in percentage

Table 1: Performance of feature mining in the English Dev set with RFC

In Table 1, we can see that by using the period punctuation as the baseline feature set, the classification performance is already decent, with 0.76 F1-Mean. And the ‘ES’ prediction is apparently more accurate than the ‘BS’ prediction, which is intuitively reasonable as a period usually marks an end of a sentence.

By adding the other feature set one by one, as mentioned in Section 3.1, the performance consistently increases with 1-3% F1 improvement. Some features help on both ‘BS’ and ‘ES’, such as ‘Punc2’, ‘Cap’, ‘Acro’, ‘Lett’, ‘POS’; some help only on ‘BS’, such as ‘Enter’; and some help only on ‘ES’, such as ‘Dig’. Among all the feature sets, POS shows the greatest improvement to the identification of both ‘BS’ and ‘ES’, which implies the usefulness of incorporating certain syntactic information into sentence detection.

4.4 Keyword Validation

This section focuses on the comparison of post-classification validation vs. non-validation of keyword knowledge to demonstrate the effectiveness of knowledge enhancement. The basic experimental setting is the same, including the fused feature set, the RFC classifier and bilingual training. As we submitted 2 runs of both languages with the RFC method for the contest,

the current comparison is based on both the validation (Dev) set and the test set. The corresponding classification results are shown in Table 2 below.

	F1	NO ^a	YES ^b	$\delta(\%)$
Dev	BS	0.83	0.83	0
	ES	0.86	0.87	1 \uparrow
	Mean	0.845	0.85	0.5 \uparrow
Test	BS	0.81	0.84	3 \uparrow
	ES	0.88	0.88	0
	Mean	0.845	0.86	1.5 \uparrow

^a Without keyword validation

^b With keyword validation

Table 2: Performance of RFC in the French Dev and Test sets in terms of keyword validation

Informative knowledge can be a very useful guidance to correcting the confused predictions of statistical models, as evidenced in this experiment. We implemented the keyword extraction and validation procedure, as shown in Section 3.3, to post-process the predictions of the RFC model with the aim of rectifying certain wrong labels caused by confusion of the out-of-boundary keywords.

The results in Table 2 indicate that keyword validation is indeed successful for both validation and testing. Notably, the improvement of predicting ‘BS’ is significant (3% \uparrow), which leads to a 1.5% F1-Mean gain for our system in the final contest, and this result is comparable to the top teams. The success of keyword validation in our experiment suggests that by using adequate domain knowledge in NLP tasks, we could optimize the classification performance in an efficient way. Moreover, the domain knowledge itself serves a valuable resource for text processing and information extraction of the specific domain.

5 Knowledge Adaptation to the Test Sets and the Final Results

This section aims to fill in the gap of our mistake in missing the implementation of the knowledge enhancement procedure on the test sets, which causes an unexpected low result of the English trial for RFC.

In order to remedy the above mistake, we simply run the script of the same procedure in Section 3.3 by including the keywords of test sets in the knowledge dictionary so as to cover the additional domain specific words that are not included/recoverable in the training data⁵. The corresponding results and ranks of our system are shown in Table 3 below.

As Table 3 shows, the results of RFC for the French trial are consistent among the three types of validation and our team achieves a stable rank (No. 8) in the competition. However, for the English trial,

⁵https://github.com/ClaraWan629/FinSBD_RFC_r1

Test Set	F1-Mean	Rank
Dev_en_rfc1	0.875	—
Test_en_rfc1	0.78*	16 [◇]
Test_en_rfc1_adapted	0.835*	10*
Dev_fr_rfc1	0.85	—
Test_fr_rfc1	0.86*	8 [◇]
Test_fr_rfc1_adapted	0.86*	8*

* Result without adaptation to the test set

◇ Rank without adaptation to the test set

* Result with adaptation to the test set

* Rank with adaptation to the test set

Table 3: Performance of RFC *w.r.t.* knowledge adaptation

there is a 9.5% F1-Mean gap between Dev_en_rfc1 and Test_en_rfc1, which is surprisingly different from our estimation. As mentioned above, we conduct a keyword adaption step on the test set and obtained a more reasonable result of the English trial with a rank of 10, as highlighted in Table 3. This adaptation step necessarily proves that our method of feature engineering and knowledge enhancement is effective and robust and it is important to resolve the over-fitting problem by applying it to the test sets.

6 Conclusion

In this work, we demonstrate the efficiency and robustness of combining feature engineering, bilingual training and knowledge-enriched approaches to the detection of sentence boundaries for noisy data in Financial NLP. We first conduct document introspection and error analysis in mining salient features for fitting the models and the final features include a 24-dimension array of punctuation, digital number, capitalization, acronym, letters, Enter, and POS tags. We then tune the classifiers on parameters of min_samples_split and max_features for RFC and batch size, epochs for NN with optimized performance on Dev sets. We also implement rule-based validation of keyword knowledge extracted from the out-of-vocabulary word sequences in FinSBD’s datasets. Lastly, we train on both English and French datasets to make predictions with maximal training data. The results of the four aspects of comparisons suggest the following findings: 1) NN does not showing significant advantage over traditional classifiers as RFC is better fitted to the selected feature set in this work; 2) NN performs better in terms of new features not originally selected; 3) The significant improvement of using POS information of a three-dimension sequence in the task shows that syntactic information may be helpful for sentence detection; 4) Informative knowledge enhancement shows a double benefit for both the correction of misclassification of statistical models and resource construction in domain-specific NLP tasks. It is important to note that our unexpectedly lower result of RFC for the English trial is found to be caused by the mistake of not

implementing knowledge enhancement on the test set. After we conduct a knowledge adaptation to the test set, the outcome achieved is close to our estimation (English: 0.835%; French: 0.86%) and within reasonable range of the best results. Although our result is not currently the best, our system is designed to be highly adaptive with minimal training data for a new language and/or a novel domain. We hope to conduct additional studies to verify the effectiveness of this feature design.

Acknowledgments

This work is partially supported by the fund of the GRF grant (RGC Ref No. 15608618).

References

- [Ait Azzi *et al.*,] Abderrahim Ait Azzi, Houda Bouamor, and Sira Ferradans. The FinSBD-2019 Shared Task: Sentence boundary detection in PDF Noisy text in the Financial Domain.
- [Breiman *et al.*, 1984] Breiman, Friedman, Olshen, and Stone. Classification and regression trees. *Wadsworth Int. Group*, 37(15):237–251, 1984.
- [Breiman, 1999] L. Breiman. Random forests—random features. *Technical Report 567*, 1999.
- [Chen *et al.*, 2009] Chen, Huang, Tian, and Qu. Feature selection for text classification with naïve bayes. *Expert Systems with Applications*, 36(3):5432–5435, 2009.
- [Garla and Brandt, 2012] Vijay N. Garla and Cynthia Brandt. Ontology-guided feature engineering for clinical text classification. *Journal of biomedical informatics*, 45(5):992–998, 2012.
- [Ghosh and NAG, 2002] Joydeep Ghosh and Arindam C. NAG. Knowledge enhancement and reuse with radial basis function networks. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN’02 (Cat. No. 02CH37290)*, pages 1322–1327. IEEE, 2002.
- [Griffis *et al.*, 2016] Griffis, Shivade, Lussier E. Fosler, and A. M. Lai. A quantitative and qualitative evaluation of sentence boundary detection for the clinical domain. In *AMIA Summits on Translational Science Proceedings*, page 88, 2016.
- [Hearst, 1994] Marti A Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 9–16. Association for Computational Linguistics, 1994.
- [Hou *et al.*, 2019] Renkui Hou, Chu-Ren Huang, and Hongchao Liu. A study on chinese register characteristics based on regression analysis and text clustering. *Corpus Linguistics and Linguistic Theory*, 15(1):1–37, 2019.
- [Huang and Chen, 2017] Chu-Ren Huang and Keh-Jiann Chen. Sinica treebank. In *Nancy Ide and James*

- Pustejovsky (Eds.), Handbook of Linguistic Annotation*, pages 641–657. Dordrecht: Springer, 2017.
- [Kiss and Strunk, 2006] Tiber Kiss and Jan Strunk. Un-supervised multilingual sentence boundary detection. *Computational linguistics*, 32(4):485–525, 2006.
- [Lee and Lee, 2006] Changki Lee and Gary G. Lee. Information gain and divergence-based feature selection for machine learning-based text categorization. *Information processing & management*, 42(1):155–165, 2006.
- [McCulloch and Pitts, 1943] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [Mikheev, 2000] Andrei Mikheev. Tagging sentence boundaries. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 264–271. Association for Computational Linguistics, 2000.
- [Mikheev, 2002] Andrei Mikheev. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):289–318, 2002.
- [Palmer and Hearst, 1997] David D. Palmer and Marti A. Hearst. Adaptive multilingual sentence boundary disambiguation. *Computational linguistics*, 23(2):241–267, April–June 1997.
- [Read, 2012] Jonathon Read. Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India, 2012. Association for Computational Linguistics.
- [Reynar and Adwait, 1997] Jeffrey C. Reynar and Ratnaparkhi Adwait. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, pages 16–19. Association for Computational Linguistics, March 1997.
- [Riley, 1989] Michael D. Riley. Some applications of tree-based modelling to speech and language. In *Proceedings of DARPA ,5'peech and Language Technology Workshop*, pages 339–352, Cape Cod, Massachusetts, 1989.

AI Blues at FinSBD Shared Task: CRF-based Sentence Boundary Detection in PDF Noisy Text in the Financial Domain

Ditty Mathew, Chinnappa Guggilla

Applied Intelligence Labs - Accenture Operations
Accenture Solutions Pvt Ltd, Bangalore, India

{ditty.mathew,chinnappa.guggilla}@accenture.com

Abstract

This paper reports the team AI.Blues's participation in the FinSBD 2019 shared Task on 'Sentence Boundary Detection in PDF Noisy Text in the Financial Domain'. Sentence detection from noisy text is a challenging task. We modeled the sentence boundary detection problem as a sequence labeling problem using Conditional Random Field (CRF) approach for English and French language financial texts. We proposed to use punctuation embeddings as an additional feature along with the basic language specific features and obtained 84.5%(F1) and 86.5%(F1) accuracies in the English and French language shared task datasets respectively.

1 Introduction

The task of Sentence Boundary Detection (SBD) is to identify the sentence segments within a text. In Natural Language Processing (NLP), the *sentence* is the foundational unit and extracting sentences or detecting the boundary of sentences from a noisy text is a challenging task. Any imperfect sentence boundary detection system can affect the morphologic, syntactic, semantic and discourse analysis in text processing. The punctuations such as '.', '?' and '!' are commonly used as sentence boundaries. However, the usage of punctuation '.' is ambiguous [Grefenstette and Tapanainen, 1994]. It can be used along with decimals, email addresses, abbreviations, initials in names, etc.

Despite the important role of sentence boundary detection in NLP, this area has not received enough attention so far. The existing approaches for this task are confined to formal texts and to the best of our knowledge no studies have been conducted in noisy texts for this task. In FinSBD shared task, the focus is to detect the beginning and ending boundaries for extracting well segmented sentences from financial texts. These financial texts are PDF documents in which investment funds precisely describe their characteristics and investment modalities. The noisy unstructured text from these PDF files was parsed by the shared task organizers and the task is to transform them into semi-structured text by tagging the sentence boundaries in two languages - English and French. For example: consider the English sentence "*Subscriptions may only be received on the basis of this Prospectus.*". Here the

word *Subscriptions* is tagged as the beginning and the period¹ '.' is tagged as the ending of the sentence in the given corpus. We have modeled the sentence boundary detection problem as a sequence labeling problem. The tokenized text is the input and the output is the corresponding labels. The labels assigned to the tokens are 'BS', 'ES' and 'O' to mark the beginning of the boundary, ending of the boundary and non-boundary token respectively.

We propose a Conditional Random Field (CRF) [Lafferty *et al.*, 2001] model to predict the label sequence of the input text. The rules for detecting sentence boundaries can be captured as features of CRF and learns the conditional probability of the label sequence given the observation sequence of features. We report the related work in Section 2 and briefly discussed the idea of conditional random field in Section 3. In Section 4, we explain the proposed part of speech and punctuation embeddings-based clustering features for the CRF model in this task. Section 5 presents the data sets, experiments, evaluation and its results. Section 6 summarizes the error analysis and discussion which is followed by conclusion and future work in Section 7.

2 Related Work

In the literature, the approaches attempted for sentence boundary detection task fall into three categories - rule-based approach, supervised machine learning approach and unsupervised approach. The rule-based SBD uses hand-crafted rules and heuristics. Mikheev [2002] proposed a rule-based approach which disambiguates the occurrence of period/full stop by determining whether it decides the sentence boundary or not. This method identifies the abbreviations by looking at local contexts and the repetitions of individual words in the document. It then applies this information to detect sentence boundary by applying a small set of rules.

Recent research in sentence boundary detection focuses on machine learning techniques. Riley [1989] presented a decision tree classifiers in determining whether the instances of full stops mark sentence boundaries. This approach uses features such as probabilities of words being sentence final or initial, word length, and word case. Satz is an approach proposed by Palmer and Hearst [1997] which uses decision tree

¹we use the term period or full stop interchangeably to refer the punctuation '.'

or a neural network to disambiguate the role of punctuation mark in a sentence by using the prior distributions of word class surrounding the possible end-of-sentence punctuation mark as features. A maximum entropy learning is proposed by Reynar and Ratnaparkhi [1997] which disambiguates the potential sentence boundary tokens such as ‘.’, ‘?’, ‘!’. This model learns the contextual features of such ambiguous punctuations by considering the token preceding and following a sentence boundary.

Kiss and Strunk [2006] proposed an unsupervised sentence boundary detection system called Punkt. This method detects abbreviations, initials and ordinal numbers by using collocation information as evidence derived from unannotated corpora. A large development corpus of the Wall Street Journal is used to derive the collocation information. This method is proposed for sentence boundary detection in multilingual sentences.

Closest to our proposed approach is the work on token and sentence splitters using conditional random field in biomedical corpus [Tomanek *et al.*, 2007]. This model captures features such as - i) token size, ii) sentence boundary tokens such as full stop, question mark, and exclamation mark, iii) canonical form of word based on the usage of capital letter, small letter, digit and other characters, iv) orthographical features such as HasDash, AllCaps, InitialCap, and hasParenthesis, and v) abbreviations. In our proposed model, we study the contextual behaviour of punctuations and formation of rules based on the combined usage of sentence boundary tokens. Evang *et al.* [2013] proposed a sentence segmentation method using CRF which considers characters as basic units for labeling. However, in FinSBD shared task, tokens are the basic units for labeling.

With respect to the sentence boundary detection of the French language, Maegaard, and Spang-Hanssen [1973] described a method to segment the French sentences into principal clauses and subordinate clauses by using only a few kinds of linguistic signs in the text. Gonzalez *et al.* [2018] proposed a convolutional neural network [Kalchbrenner *et al.*, 2014] based approach for detecting sentence boundaries of French speech texts which tackle the task as a binary classification task.

3 Conditional Random Field

Conditional Random Field (CRF) [Lafferty *et al.*, 2001] is a probabilistic method for structured prediction and it computes the conditional probability of the label sequence given the observation sequence. The conditional probability of the label sequence $Y = y_1, y_2, \dots, y_T$ given the observation sequence $X = x_1, x_2, \dots, x_T$ is given as

$$P(Y/X) = \frac{1}{Z(X)} \exp \sum_{t=1}^T \sum_{k=1}^F \lambda_k f_k(x_t, y_t) \quad (1)$$

where $f_k(x_t, y_t)$ is a feature function and its value may range from $-\infty$ to $+\infty$, but typically they are binary. Each feature function f_k is associated with a weight λ_k which is learned during training. $Z(X)$ is the normalization factor to make the

probabilities sum up to 1 and it is defined as

$$Z(X) = \sum_Y \exp \sum_{t=1}^T \sum_{k=1}^F \lambda_k f_k(x_t, y_t) \quad (2)$$

The conditional distribution discussed in Equation 1 is a linear chain CRF which includes the features only for current word. We used richer features of the input x_i such as prefixes (x_{i-1}, x_{i-2}), suffixes (x_{i+1}, x_{i+2}) and surrounding words of current word and their corresponding label sequences.

We have modeled the sentence boundary detection problem as a sequence tagging problem. When applying CRF to SBD problem, a sequence of tokens in a text is considered as the observation sequence and the label sequence is the corresponding sequence of labels. Each token is labeled with respect to its position in the sentence. If the token is positioned at the beginning of a sentence, its label is ‘BS’. If the token is positioned at the end of the sentence, then the label is ‘ES’. The remaining tokens are labeled as ‘O’. In this way we used the English and French training corpus tagged with 3-tags for building the sentence boundary detection model using CRF.

4 Sequence Representation and Features for CRF

4.1 Preprocessing - Tokenization

The input text to the CRF model is tokenized in a way that the punctuations are considered as tokens. For example, consider the following text and its tokens from English data.

Text: GAM Star (Lux) Prospectus

Tokens: [‘GAM’, ‘Star’, ‘(’, ‘Lux’, ‘)’, ‘Prospectus’]

4.2 Sequence Representation

We consider an average of five sentences as a single unit for sequence representation in CRF modeling. The optimal size for the sequence representation is determined using the best performance of the CRF model on the development set during the training phase. In the case of test set, we have used the entire document as a sequence for CRF prediction. We have applied the same schema for both the English and French language SBD tasks.

4.3 Basic Features

In CRF, each token is represented by a set of features. In addition, the features of k preceding and k following tokens as n -grams are included for each token. The feature selection plays a crucial role in CRF. We propose the following basic set of surface and orthographic features for sentence boundary detection task. We also used part of speech (POS) syntactic feature in addition to other features and denoted as ‘basic features’ in the rest of the sections.

- **Token:** The token itself is considered as a feature. This feature captures the co-occurrence properties of tokens when we consider the preceding and following tokens.
- **Length of token:** The length of the token is considered as a feature.
- **IsUpper:** This is a binary feature which is set to 1 if all characters of the token are in upper case otherwise it is set to 0.

- **IsLower:** This is a binary feature which is set to 1 if all characters of the token are in lower case else 0. This feature is based on the assumption that a sentence may not start with a lower case character word.
- **IsTitle:** This is a binary feature which is set to 1 if the first character of the token is in upper case and the remaining characters are in lower case.
- **PosTag:** The part of speech tag of the token is used as a feature. This feature captures the role of parts of speech such as verb, noun, prepositions, etc in determining the sentence boundaries.
- **Token Name:** This feature assigns a name to the token based on its nature such as whether it is a word, punctuation, digit, etc. The assignments of token names for the corresponding token types are given in Table 1. This feature captures the characteristics of tokens.

Token type	Token Name
Word	NN
Punctuation	<Name of the punctuation>
Digit	NUMBER
Roman Numeral	ROMAN
Alphabet	ALPHABET
If token has number and character	ALPHANUMERIC

Table 1: Token names based on token type

- **Lexical combinations:** We consider features which check for the combined usage of tokens in the sentence boundaries. These features are listed in Table 2. These features obtain the contextual features of the potential sentence boundary tokens such as ‘.’, ‘?’, ‘!’ by considering the token preceding and following a sentence boundary. In table 2, we list the features only for the token ‘.’ as the financial corpus given in this shared task does not use ‘?’ and ‘!’ as sentence boundary. These features are binary features which are set to 1 if the pattern occurs for the token t_i .

$t_i = \text{'.'}, t_{i+1} = \text{A word begin with upper case}$
$t_i = \text{'.'}, t_{i+1} = \text{A word begin with upper case}, t_{i+2} = \text{A word begin with upper case}$
$t_i = \text{'.'}, t_{i+1} = \text{'('}, t_{i+2} = \text{digit/alphabet/roman numeral}, t_{i+3} = \text{'')}'$
$t_{i+4} = \text{A word begin with upper case}$
$t_i = \text{'.'}, t_{i+1} = \text{digit/alphabet/roman numeral}, t_{i+2} = \text{'')}' \text{ or } \text{'.'}, t_{i+3} = \text{A word begin with upper case}$
t_{i-1} is a word begins with upper case $t_i = \text{'.'}, t_{i+1} = \text{'('}, t_{i+2} = \text{digit/alphabet/roman numeral}, t_{i+3} = \text{'')}'$ $t_{i+4} = \text{A word begin with upper case}$

Table 2: Lexical combinations as features for CRF

4.4 Punctuation Embeddings as Clustering Feature

The word embeddings [Mikolov *et al.*, 2013a] have been proved effective in capturing contextual features and linguis-

tic regularities [Mikolov *et al.*, 2013b]. After analyzing the corpus, we observed that the punctuation collocations in the sentences would contribute to identify the sentence boundaries. For example, in a sentence if ‘-’, ‘;’ and ‘.’ occur in combination with the other content words, we could say that all these punctuations together can help in identifying the right boundary of the sentence. Since punctuations are important in deciding the sentence boundaries, we use the information in punctuation embeddings as a feature. As the embedded vector is a high dimensional vector, we cannot directly use word embeddings as a CRF feature. Hence, we represent the punctuations using embedded vectors and cluster the embedded vectors of punctuations using k -means clustering algorithm [Hartigan and Wong, 1979]. The punctuations in each cluster are assigned a distinct value based on its cluster assignment and this value is used as the feature. The tokens which are not punctuations are grouped into a different cluster. We use pre-trained glove embeddings [Pennington *et al.*, 2014] and fastText embeddings [Grave *et al.*, 2018] for English and French texts respectively.

5 Experiments and Results

In this section, we describe the data sets, features used in the CRF method and evaluation of results.

5.1 Datasets

In FinSBD shared task, data sets are provided for English and French language [Ait Azzi *et al.*, 2019]. As the corpus is related to finance domain, the text contains various data elements such as formatting indicators, titles, subtitles, sections. Each of these elements, in turn, contains various types of vocabulary including special symbols, numerals, currencies, named entities. The data sets are in JSON format which contains i) the text to detect sentence boundaries and this text is already tokenized using NLTK, ii) begin_sentence which contains the indexes of tokens in the text that mark the beginning of well-formed sentences in the text, iii) end_sentence which contains the indexes of tokens in the text that mark the end of well-formed sentences in the text. The dataset statistics for English and French languages are given in Table 3. The av-

Language	Dataset	No of tokens	No of sentences
English	Train	904,057	22,342
	Dev	49,859	1,384
	Test	56,952,	1,265
French	Train	827,852	22,636
	Dev	119,008	3,141
	Test	106,577	2,981

Table 3: Dataset Statistics

erage sentence lengths of English text in train, development, and test data sets are 30.82, 31.27 and 35.32 respectively; and that of French text are 26.71, 28.54 and 26.49 respectively. In particular to the FinSBD shared task dataset, we observed that i) the headings, bullet and numbering points etc. are considered as sentences, ii) some non-boundary tokens in sentences

begin with upper case letter, iii) some tokens are fully given in upper case, iv) the punctuation ‘-’ is used as the beginning token of bullet points more frequently.

5.2 Experiment Setup

We extract the basic features and punctuation cluster features of tokens as discussed in Section 4. The parts of speech tag is tagged using python NLTK² package for English and stanford pos tagger³ is used for French. To obtain the punctuation embeddings for English text, we use pre-trained glove embeddings [Pennington *et al.*, 2014] of 100 dimension and clustered them using k -means clustering. We experimented with different values of k ranges from 2 to 10 over the development data. We observed that the punctuations are clustered based on its contextual behaviour and the clusters resulted when $k=7$ are given in Table 4. For French, we use pretrained

Cluster No	Punctuations
1	, - . ’ : & ’ ; \$
2	()
3	[]
4	? !
5	< >
6	+ * =
7	# @ — / %

Table 4: Punctuation Clusters obtained from English text

Cluster No	Punctuations
1	” ’
2	() [] . , ; :
3	? !
4	< >
5	+ * = - /
6	# @ — &
7	\$ %

Table 5: Punctuation Clusters obtained from French text

fastText embeddings [Grave *et al.*, 2018] of 300 dimension which is trained on Wikipedia data to obtain the punctuation clusters. The clusters are listed in Table 5.

Our CRF model is compared with the baselines such as punkt sentence tokenizer [Kiss and Strunk, 2006] and sentence boundary detector proposed by Tomanek et al [2007]. The punkt tokenizer divides a text into a list of sentences by using an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences. We use the punkt tokenizer model implemented in NLTK for English and French languages to report the results. The sentence boundary detector by Tomanek et al. [2007] is a conditional random field model with the following set of features.

- The token and its length.
- A binary feature which checks whether the token is a sentence boundary symbols such as full stop, question mark, and exclamation mark.
- Canonical word form which is constructed by applying the transformation rules such as i) replace capital letters by ‘A’, ii) replace lower case letters by ‘a’, iii) replace digits by ‘0’ and, iv) replace all other characters by ‘-’.
- Features such as HasDash, AllCaps, InitalCap, has-Parenthesis.
- A binary feature which is set to 1 if the token is contained in a list of abbreviations.
- Local context features of neighboring tokens in the window [-1,1]

²<https://www.nltk.org/>

³<https://nlp.stanford.edu/software/tagger.html>

5.3 Experiments

We use python CRFsuite [Korobov and Peng, 2014] to model a linear chain CRF and it is trained using the gradient descent algorithm. The parameters such as “all_possible_transitions” and “all_possible_states” are set as True. We tuned the regularization parameters using separate development set for each model. The context window k to fetch the features of surrounding prefix and suffix tokens is selected based on the F1-score over the development data. The value k is chosen for English and French is 6. We use a paragraph as a sequence for training and the entire text in the test data as a single sequence for testing. As the prediction of boundary tokens require the characteristics of preceding and following tokens, the paragraph is considered over sentence as a sequence in the training phase. The text in the test data is input as a sequence as the paragraph information of test data is not available. During training, the sentences are constructed using the beginning and ending information given in the train data and a paragraph is considered as five consecutive sentences. The number of sentences in the paragraph is chosen based on the F1-score over the development data.

Language	Evaluation Measure	Label		Average
		BS	ES	
English	Precision	90	93	91.5
	Recall	85	90	87.5
	F1-Score	87	92	89.5
French	Precision	88	89	88.5
	Recall	86	90	88
	F1-Score	87	89	88

Table 6: Results obtained for the development data for the submitted model

Language	Evaluation Measure	Label		Average
		BS	ES	
English	Precision	77	83	80
	Recall	88	92	90
	F1-Score	82	87	84.5
French	Precision	89	90	89.5
	Recall	80	85	82.5
	F1-Score	85	88	86.5

Table 7: Results submitted to FinSBD shared task for the test data

5.4 Evaluation and Results

F1-score is used as the evaluation measure in the FinSBD shared task and we also report the precision and recall for evaluation. The precision, recall and F1-score are averaged for ‘BS’ and ‘ES’ labels and this average score is used for reporting the best model. The precision, recall, and F1-score obtained for English and French text over the development data are given in Table 6. The average F1-score obtained for English text is 89.5% and that of French text is 88% over the development data. The predicted results on the given test set are reported in Table 7. The submitted results are predicted using the CRF model which is trained using all the features discussed in Section 4. However, we later figured out that we used only a subset of all the features in the feature preprocessing step of the prediction module that we used for generating

Method	Evaluation Measure	Label		Average
		BS	ES	
Punkt [Kiss and Strunk, 2006]	Precision	58	73	65.5
	Recall	71	89	80
	F1-score	64	81	72.5
CRF [Tomanek <i>et al.</i> , 2007]	Precision	77	81	79
	Recall	83	89	86
	F1-score	80	85	82.5
Basic Features	Precision	82	84	83
	Recall	85	94	89.5
	F1-Score	83	89	86
Basic + Punctuation Cluster	Precision	82	84	83
	Recall	86	94	90
	F1-Score	84	89	86.5

Table 8: Evaluation of English gold standard test set with all the features (corrected results post the shared task submission)

results on given test data. We fixed this mistake later and reported the corrected results in Tables 8 and 9. The highest accuracy values for precision, recall, and averaged F1 measures are specified in bold font as shown in Tables 8 and 9.

In Table 8, we report the evaluation scores of baselines such as punkt [Kiss and Strunk, 2006] and CRF model [Tomanek *et al.*, 2007], and our proposed CRF models using basic features and basic + punctuation-based cluster features (See Section 4) for the English gold standard test set given in FinSBD shared task. We can observe that the CRF model using basic and punctuation-based cluster features are performing better than all other methods and, this model scores the highest F1-score for both ‘BS’ and ‘ES’ labels. The CRF model which uses basic features performs better than other baselines and scores the highest F1-score for ‘BS’. While the Punkt unsupervised model scores an F1-score of 81% for ‘ES’ label, the sentence beginning performing very poorly. All CRF based models report greater than 80% F1-score for the ‘BS’ labels and it indicates that the sequential labeling of tokens gains more information on sentence beginning. We performed a paired t-test to check the statistical significance of the improvements of proposed CRF models over the baselines [Tomanek *et al.*, 2007; Kiss and Strunk, 2006] and observed that the improvements are statistically significant with p -value less than 0.05.

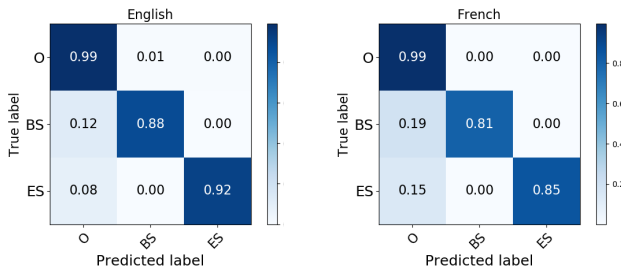


Figure 1: Confusion matrices obtained from the submitted results in the FinSBD shared task

The evaluation of French gold standard test data is re-

Method	Evaluation Measure	Label		Average
		BS	ES	
Punkt [Kiss and Strunk, 2006]	Precision	54	39	46.5
	Recall	61	82	71.5
	F1-score	57	53	55
CRF [Tomanek <i>et al.</i> , 2007]	Precision	77	81	79
	Recall	82	87	82.5
	F1-score	80	84	82
Basic Features	Precision	89	90	89.5
	Recall	87	90	88.5
	F1-Score	88	90	89
Basic + Punctuation Cluster	Precision	90	92	91
	Recall	88	90	89
	F1-Score	89	91	90

Table 9: Evaluation of French gold standard test set with all the features (corrected results post the shared task submission)

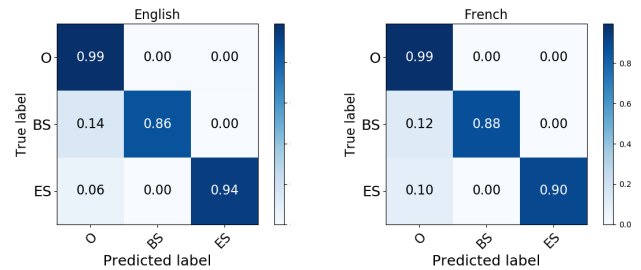


Figure 2: Confusion matrices obtained from post-submission results

ported in Table 9. The proposed CRF models are compared with the baselines and the CRF model which uses basic features. The CRF model which uses basic and punctuation cluster features is performing better than all other models in the gold standard test data. This model identifies beginning and ending of the sentence with F1-scores 89% and 91% respectively. The CRF model which uses basic features also performs much better than baselines. The CRF model by Tomanek [2007] identifies the ‘BS’ and ‘ES’ labels with F1-scores 80% and 84% respectively. The punkt model performs very poorly for French text in identifying both beginning and ending of the sentences. The improvements of the proposed models over the baselines [Tomanek *et al.*, 2007; Kiss and Strunk, 2006] are statistically significant with p -value less than 0.05 in paired t-test.

6 Error Analysis and Discussion

We have computed the confusion matrices on the shared task test results and for the post shared task submission results for English and French SBD tasks for the 3 tags - ‘BS’, ‘ES’ and ‘O’ tags. These confusion matrices are illustrated in Figures 1 and 2 for English and French data sets respectively. These confusion matrices are actually normalized by its values to avoid the skewed ‘O’ tag distortion for compact presentation of matrices.

In English SBD task, the average precision of ‘BS’ and ‘ES’ tags is relatively less when compared to the average re-

call of these tags as these tags are mostly confused with ‘O’ tag as shown in Figure 1. We also observe that the ‘BS’ and ‘ES’ tags in English shared task are not confused with each other as they were with ‘O’ tag. Same is seen in the confusion matrices of the improved post shared task submission results as shown in Figure 2.

In the case of French SBD task, the average precision of ‘BS’ and ‘ES’ tags is higher when compared to the average recall as these tags were mostly confused with ‘O’ tag as shown in the Figure 1. We also observe a negligibly small number of cases, ‘BS’ and ‘ES’ tags were got confused. The same behavior could be observed in the improved results of post-shared task submission as shown in Figure 2.

6.1 Error analysis of English text

In the case of English SBD task, we manually examined those sentences that are misclassified as ‘O’ (45% of the total errors) and found that they tend to contain short sentences, sentences starting with lower case words, hyphens, bullet points, named entity tokens (for example: GAM Star); numbering/currency tokens with brackets (for example, (a), a), etc.). Following are some examples of these cases.

- includes but is not limited to:
- – of issues linked to “emerging-country risks”.
- a) to e) are raised to a maximum of 20% for investments in Shares
- The European Union.

Some example sentences where the end tag ‘ES’ is predicted as ‘O’ tag.

- Cash collateral received may only be
- – of issuers domiciled in emerging countries , or
- The minimum capital is equivalent in US Dollar to EUR 1,250,000.00.
- available to distributors who have entered into arrangements with the GAM Group.

We also observed a few sets of errors in the ground truth test data especially when sentences start with bullet points and having currency numbers and other punctuation symbols. In rest of the 55% errors where the ‘O’ tag is predicted as ‘BS’ and ‘ES’, it is found that punctuations such as ‘-’, ‘:’ and ‘.’ inside the title, as non-boundary tokens are misclassified as ‘ES’ and the tokens followed by such non-boundary tokens are also misclassified as ‘BS’ label.

6.2 Error analysis of French text

In case of French SBD task, we manually examined those sentences that are misclassified as ‘O’ (65% of the total errors) and observed that the short sentences occurring in the title, bullet points, structured segments containing currency numbers, etc are attributed to major errors. The following are some sentences for the prediction error where the beginning token is predicted as ‘O’ tag.

- 52,45 Euros.
- * le solde, s’il existe, est réparti entre les Parts A et B comme suit:

- CAMGESTION, une société de gestion appartenant au groupe BNP Paribas.
- (a) une Personne non Eligible et,

Examples of sentences where the ending token is predicted as ‘O’ tag are given below.

- Le FCP est exposé, entre:
- Fonds commun de placement de droit français (FCP)
- Siège social : 1, boulevard Haussmann - Paris 75009
- Tous les jours ouvrés jusqu ’ à 11:00, heure de Paris.

The first sentence end with ‘:’, second and third sentences are isolated sentences and they are not ending with fullstop.

The sentences where both beginning and ending tokens are predicted as ‘O’ are mostly titles and short sentences. Some example sentences are given below.

- FIA soumis au droit français
- Éligibilité : PEA
- Néant.
- Parts G : 300000€

In rest of the 35% errors where the ‘O’ tag is predicted as ‘BS’ and ‘ES’, it is found that punctuations such as ‘-’, ‘:’ and ‘.’ inside the title, as non-boundary tokens are misclassified as ‘ES’ and the tokens followed by such non-boundary tokens are also misclassified as ‘BS’ label.

Same pattern of errors are observed in the results of post shared task submission as shown in tables 8 and 9. Properly handling these type of sentences would require modeling with complex features such as combinations of punctuation, indentation and formatting indicators, currency symbols with nonlinear chain CRFs and advanced deep sequential neural networks such as bi-directional LSTMs.

7 Conclusion and Future Work

We presented the experimental results of FinSBD Shared Task: CRF-based Sentence Boundary Detection in PDF Noisy Text in the Financial Domain. There are 2 tasks for English and French financial texts. We modeled SBD as a sequential modeling approach and obtained 84.5% (F1) on English data set and 86.5%(F1) in French task using basic features in combination with punctuation-based embeddings and syntactic POS tags. After correcting the bug in the prediction code, we actually observed 86.5% (F1) and 90%(F1) in English and French SBD tasks respectively.

Sentence boundary detection in noisy pdf texts poses challenges, such as working with semi-structured text containing format indicators such as bullets, numerals, financial numbers and specialized vocabularies such as named entities. One of the future directions is to explore proximity specific meta structural features with better sequence representation in the dynamic CRFs to capture long range dependencies. Experimenting with hybrid CRF and bi-directional long short-term deep memory networks would also be our future work for getting the improved results.

References

- [Ait Azzi *et al.*, 2019] Abderrahim Ait Azzi, Houa Bouamor, and Sira Ferradans. The FinSBD-2019 Shared Task: Sentence boundary detection in PDF Noisy text in the Financial Domain. In *The First Workshop on Financial Technology and Natural Language Processing (FinNLP 2019)*, Macao, China, 2019.
- [Evang *et al.*, 2013] Kilian Evang, Valerio Basile, Grzegorz Chrupała, and Johan Bos. Elephant: Sequence Labeling for Word and Sentence Segmentation. In *EMNLP 2013*, 2013.
- [González-Gallardo and Torres-Moreno, 2018] Carlos-Emiliano González-Gallardo and Juan-Manuel Torres-Moreno. Sentence Boundary Detection for French with Subword-level Information Vectors and Convolutional Neural Networks. *arXiv preprint arXiv:1802.04559*, 2018.
- [Grave *et al.*, 2018] Edouard Grave, Piotr Bojanowski, Prakhara Gupta, Armand Joulin, and Tomas Mikolov. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [Grefenstette and Tapanainen, 1994] Gregory Grefenstette and Pasi Tapanainen. What is a Word, What is a Sentence?: Problems of Tokenisation. 1994.
- [Hartigan and Wong, 1979] John A Hartigan and Manchek A Wong. Algorithm AS 136: A K-means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.
- [Kalchbrenner *et al.*, 2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A Convolutional Neural Network for Modelling Sentences. *arXiv preprint arXiv:1404.2188*, 2014.
- [Kiss and Strunk, 2006] Tibor Kiss and Jan Strunk. Un-supervised Multilingual Sentence Boundary Detection. *Computational Linguistics*, 32(4):485–525, 2006.
- [Korobov and Peng, 2014] M Korobov and T Peng. Python-crfsuite, 2014.
- [Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, 2001.
- [Maegaard and Spang-Hanssen, 1973] Bente Maegaard and Ebbe Spang-Hanssen. Segmentation of French Sentences. In *COLING 1973 Volume 2: Computational And Mathematical Linguistics: Proceedings of the International Conference on Computational Linguistics*, volume 2, 1973.
- [Mikheev, 2002] Andrei Mikheev. Periods, Capitalized Words, etc. *Computational Linguistics*, 28(3):289–318, 2002.
- [Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013.
- [Mikolov *et al.*, 2013b] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.
- [Palmer and Hearst, 1997] David D Palmer and Marti A Hearst. Adaptive Multilingual Sentence Boundary Disambiguation. *Computational Linguistics*, 23(2):241–267, 1997.
- [Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- [Reynar and Ratnaparkhi, 1997] Jeffrey C Reynar and Adwait Ratnaparkhi. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19. Association for Computational Linguistics, 1997.
- [Riley, 1989] Michael D Riley. Some Applications of Tree-based Modelling to Speech and Language. In *Proceedings of the Workshop on Speech and Natural Language*, pages 339–352. Association for Computational Linguistics, 1989.
- [Tomanek *et al.*, 2007] Katrin Tomanek, Joachim Wermter, and Udo Hahn. Sentence and Token Splitting based on Conditional Random Fields. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, volume 49, page 57, 2007.