Feature Fusion with Hand-crafted and Transfer Learning Embeddings for Cause-Effect Relation Extraction

Abdul Aziz[a], Afrin Sultana[a], MD. Akram Hossain[a], Nabila Ayman[a] and Abu Nowshed Chy[a]

[a]*Department of Computer Science & Engineering, University of Chittagong, Chattogram-4331, Bangladesh*

Abstract

Cause-effect relation extraction is the problem of detecting causal relations expressed in a text. The extraction of causal-relations from texts might be beneficial for the improvement of various natural language processing (NLP) tasks including Q/A, text-summarization, opinion mining, and event analysis. However, cause-effect relation in the text is sparse, ambiguous, sometimes implicit, and has a linguistically complex construct. To address these challenges FIRE-2020 introduced a shared task focusing on cause-effect relation extraction (CEREX). We propose a feature based supervised classification model with a naive rule-based classifier. We define a set of rules based on a causal connective dictionary and stop-words. Besides, we use a fusion of hand-crafted features and transfer learning embeddings to train our SVM based supervised classification model. Experimental results exhibit that our proposed method achieved the topnotch performance for cause-effect relation extraction and causal word annotation.

Keywords

cause-effect relation, hand-crafted features, sentence embedding, word embedding, features fusion

1. INTRODUCTION

Nowadays we see the exponential growth of web traffic, which makes a vast amount of unstructured web data. The cause-effect relation extraction (CEREX) from text becomes one of the most prominent fields in computational linguistics [1]. The concept of causality is a relationship between two events *e1* and *e2* is that occurrence of *e1* results in the occurrence of *e2*. For example, *"The accident caused a major traffic snarl on the arterial road"* - the event *"accident"* is causing the event *"snarl"*.

Automatic identification of causal-relations in texts is very important for various NLP applications including question-answering (Q/A), document-summarization, opinion mining, event analysis, product recommendation, and information retrieval. Many organizations that are specializing in web intelligence and knowledge graph creation - provide services for the analysis and representation of vast amounts of textual information. Therefore, the extraction of causal-relation might help them to create new insights into their services.

Lexicon-syntactic patterns may represent a causal relation explicitly. Besides, causality is easily understandable while expressed using different propositions (such as passive), causal links, causative verbs, causation adjectives, adverbs, and conditionals. However, there are a huge number of cases that can evoke a causal relation, but not uniquely. Therefore, the automatic extraction and identification of causal-relation in the text has become a challenging task.

FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India

aziz.abdul.cu@gmail.com (A. Aziz); afrin.sultana.cu@gmail.com (A. Sultana); akram.hossain.cse.cu@gmail.com (MD.A. Hossain); nabila.aymun.cu@gmail.com (N. Ayman); nowshed@cu.ac.bd (A.N. Chy)

The first four authors have equal contributions.

Table 1
Example of Task A.

Sentence Examples	Label
S#1: Procedure to drain resulted in restricted duty.	Causal
S#2 : There are many things needing improvement.	Non-causal

Table 2
Example of Task B.

Sentence Examples	Tagging
S#1: Heavy rains during the harvest may lead to the rotting of onion production.	Heavy\Cause rains\Cause during the harvest may\causal connective lead\causal connective to\causal connective the rotting \Effect of\Effect onion\Effect production\Effect.

To address the challenges of cause-effect relation extraction in texts, Sinha et al. proposed a shared task at FIRE-2020. The task is divided into two subtasks. In task A, a system needs to determine whether a given text contains a causal event, whereas in task B, a system needs to annotate each word in a text in terms of the four labels including cause (C), effect (E), causal connectives (CC), and none. To elucidate the definition of both tasks A and B, we articulate a few examples in Table 1 and Table 2, respectively.

The major contribution of this paper is that we proposed a feature based supervised classification framework with a naive rule-based classifier. We define a set of rules based on causal connectives and stop-words. For feature extraction, we extract a rich set of hand-crafted features by exploiting causal patterns, parts-of-speech information, lexical, and textual syntaxes and patterns. Besides, we exploit the pretrained sentence and word embeddings to extract effective transfer learning features. Experimental results elucidate that our model achieved the topnotch performance to tackle the cause-effect relation extraction task.

We organize the rest of the paper as follows: Section 2 presents the discussion of notable related work, whereas in Section 3 we describe the various components of our proposed framework. In Section 4, we present our experimental settings and analyze the performance of our model against the various settings and related methods. Finally, we conclude our paper in Section 5 with some future directions.

2. RELATED WORK

Prior works on exploring causal relations in texts are not relatively abundant. Girju et al. [2] focused on the "noun-verb-noun" pattern to detect causality in texts. Later, they used causation patterns and ambiguous verbal lexico-syntactic patterns referring to causation to detect causal relations for Q/A [3]. Rink et al. [4] extracted graph patterns to capture the contextual information of a pair of events, whereas Hidey and McKeown [5] presented a distant supervision method for causality identification by creating a new corpus from Wikipedia for causality and extracting a subset of relations with AltLexes. Recently, Dasgupta et al. [1] used bidirectional LSTM with additional linguistic feature embeddings and word embeddings for extracting causal relations in texts.

http://fire.irsi.res.in/fire/2020/home
An open class of markers in the Penn discourse tree bank (PDTB) [6]

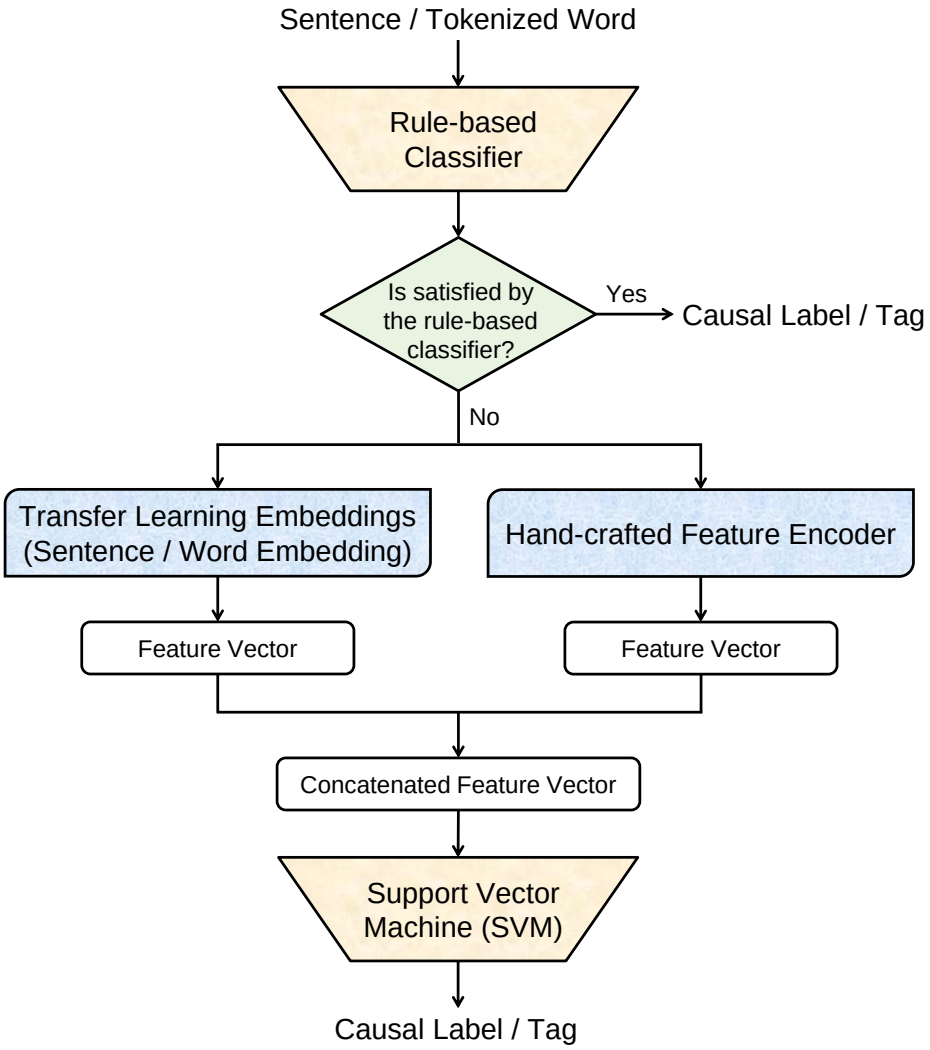

Figure 1: Overview of our proposed framework.

3. PROPOSED FRAMEWORK

We describe our proposed framework for Task A and Task B in this section. Given a sentence, in task A, we aim to categorize it into causal or non-causal, whereas in task B, we annotate each word of the sentence as a cause (C), effect (E), causal connectives (CC), and none. The overview of our proposed framework for both tasks is shown in Figure 1.

At first, we perform preprocessing for a given sentence. Our preprocessing method includes lexical normalization, expand short words, convert emoji into text, special-character removal, URL removal, consecutive words removal, and punctuation especially full-stop flooding removal. After preprocessing, we employ a rule-based classifier that classifies the sentence (for task A) / tokenized word (for task B) to the corresponding label. Sentence/word, which is not classified by the rule-based classifier, we extract a set of hand-crafted features and embedding features from a pretrained sentence (for task A) / word (for task B) embedding model. Based on the extracted features, an SVM classifier is then employed to classify/annotate the sentence/word to the corresponding label.

3.1. Data Preprocessing

We employ various preprocessing methods on the given data. We expand the contraction (e.g. "can't", "isn't", and "aren't") into their normal form for effective representation. Since the special characters and URLs do not contain any causal indicative information, we discard them from the sentences. We employ a publicly available python library emot to demojize (i.e. convert emojis into text) all the available emoji in the sentences. Besides, all the characters, words, and punctuation floodings are replaced with a single one. For example, *"This app is very very very very useful for kids...."* is becoming *"This app is very useful for kids."* after removing consecutive words and punctuation flooding. A sentence may contain some non-standard word form e.g. *"plz"* in place of *"please"*, *"Thnx"* in place of *"Thanks"*, and *"vry"* in place of *"very"*. To normalize such words, we follow a similar kind of approach used in Ref. [7] where they utilized two normalization dictionaries.

3.2. Rule-based Classifier

Rule-based classifiers are very popular in various classification tasks due to its simple design, easily explainable, effective, and quick classification performances [8, 9, 10]. In a rule-based classifier, we usually define a set of rules that estimate a certain combination of patterns most likely related to the different labels [11].

For Task A, we used a rule-based classifier where we define a rule to label a sentence as causal on the presence of causal connectives. For example, consider a sample sentence, *"I couldn't get the extra 2 hours check out due to full occupancy"*. Here, *"due to"* is a causal connective and this sentence will label as causal. We survey the related work [12] and various sources to create a causal connective dictionary that contains 121 causal connectives.

For Task B, we use two rules. At first, we check whether there is a causal connective available in the given sentence or not based on the above-mentioned dictionary. If causal connectives are present in the text, we labeled all of its tokens as CC. Next, in the rest of the sentence, we check whether there is any stopword available based on the NLTK's standard stoplist. If any stopwords available in the sentence, we labeled them as None.

3.3. Feature Extraction

We employ the hand-crafted features and transfer learning embeddings based features for the effective representation of each sentence. We exploit various lexical, syntactic, parts-of-speech (POS), and causal indicators to extract a rich set of hand-crafted features. To extract the effective transfer learning embedding based features, we utilize the pre-trained sentence embedding model InferSent [13] for Task A and word embedding model fastText [14] for Task B.

3.3.1. Hand-Crafted Features

The bag-of-words (BoW) representation of sentences uses the word occurrence statistics from the given training samples. In or BoW based n-gram features, we employ the n-gram range (1,2) along with the TF-IDF weighting scheme. Besides, we extract a rich set of hand-crafted features to exploit the cause-effect relation in the sentences. Some extracted features are common for both task A and task B and some are specific for each task. The definitions of our extracted features are described in Table 3. There are 13 features that are common for both tasks and we extract the 16 and 17 specific features for task A and task B, respectively.

https://github.com/NeelShah18/emot

Table 3
Definition of hand-crafted features used in our work.

Sl.	Feature Name	Feature Description
Common features for both Task A and Task B.		
1.	Causal Verb Presence [15]	Check if a text contains any causative verb.
2.	Simple Causal Verb Count [15]	Count simple causal verb in a text.
3.	ModAux Count [16]	Number of modal auxiliary verbs contains in a text.
4.	Causal Preposition Count [15]	The number of causal prepositions in a text.
5.-12.	POS Features [16]	Count POS in a text including proper noun, noun, pronoun, personal pronoun, adjective, verb, adverb, and conjunction.
13.	Negative polarity Count	Check whether a text has negative polarity or not.
Other features for Task A.		
14.	Average Word Length	The average word length of a text.
15.-20.	POS Percentage Features [11]	Percentage of various POS in a text including noun, pronoun, adjective, verb, adverb, and conjunction .
21.	CDN Check [16]	Whether a text contains cardinal number or not.
22.	Positive Polarity Check	Check whether a text has positive polarity or not.
23.	Is Passive Voice [15]	Check if a text contains passive voice.
24.	Subordinate Clause [16]	Check whether a comma separated sentence available in a text.
25.	Vp_with_adv Count [17]	Check if a sentence contains "Verbal Phrase + Adverb" pattern.
26.	Causal Phrase Presence [12]	Check if a sentence contains causal adjective or adverbial or prepositional Phrase.
27.	Causal Verb Count [16]	The number of causative verb present in a text.
28.	Polarity Count	Extract polarity score of the sentence.
29.	Time ML [18]	Check Presence of Time- Expression
Other features for Task B.		
14.-19.	Pattern based Features	Check whether a word is determiner, subordinate conjunction, coordinating conjunction, comparative adjectives, superlative adjectives, and possessive pronoun or not. (Binary feature.)
20.	VBGcount [16]	Count gerund or present participle (v+ing) verbs.
21.-29.	Tag based Features [1]	Check whether a word is an entity, namely, state, event, phenomenon, group, act, possession, and interjection tag or not. (Binary feature.)
30.	Temporal Feature [18]	Check if a word is temporal indicator word e.g. "end, begin, before, and after."

To estimate the different types of POS features and POS percentage features, we use the publicly available Stanza [19] toolkit. All the sentiment polarity based features are extracted using TextBlob [20] API. Besides, a publicly available python TimeX library is used to estimate the temporal features and we use our generated lexicons to estimate the causal verb and preposition related features. The rest of the linguistic features are estimated using SpaCy [21] and NLTK wordnet [22].

3.3.2. Transfer Learning Embeddings

An embedding is a transformation of a high-dimensional vector into a low-dimensional space that preserves the information in its features. To extract the effective transfer learning embedding features, we exploit the pretrained sentence and word embedding models without fine-tuning for task A and task B, respectively. We use the pretrained InferSent [13] model for sentence embedding and pretrained fastText [14] model for word embedding.

InferSent [13] is a universal sentence embedding model trained using the supervised data of the Stanford natural language inference (SNLI) datasets. We used the InferSent model trained on fastText [14] vectors to encode each sentence into a 4096-dimensional feature vector.

FastText [14] is a popular word embeddings technique that uses the state-of-the-art character-level approach to embeddings. It represents each word is modeled by a sum of vectors, with each vector representing an n-gram. We use the fastText to encode each word into a 300-dimensional feature vector.

3.4. Fusion of Hand-crafted and Transfer Learning Embedding Features

Upon extracting different kinds of features, we concatenate them to generate a unified feature vector. For Task A, we extract n-gram features, 29 hand-crafted features, and 4096-dimensional transfer learning embedding features based on InferSent [13]. Then, we combine these features to generate a unified feature vector. For Task B, we extract n-gram features, 30 hand-crafted features, and 300-dimensional transfer learning embedding features based on fastText [14]. Then, we combine these features to generate a unified feature vector.

3.5. Classification Model

At first, we employ the rule-based classifier (RBC) to classify the sentence/word to the corresponding label. For the sentence/words that are not classified by the rule-based classifier (RBC), we employ the Scikit-learn [23] implementation of the SVM model with the linear kernel (LinearSVC) where the hand-crafted and transfer learning embedding based unified feature vector is used to train and classify [24]. SVM algorithm generally works by defining a hyperplane on N-dimension and handle both the linear and non-linear characteristics of the data effectively.

4. EXPERIMENTS AND EVALUATION

4.1. Dataset Collection and Evaluation Measures

Automatic cause-effect relationship extraction from text (CEREX) task at FIRE-2020 provided a benchmark dataset to evaluate the performance of the participant's proposed systems. The training set for task A and task B consisted of 5999 texts and the portion of the text related to cause and effect also provided. If a text does not contain both the cause and effect portion then the text is labeled as non-causal and causal otherwise. The organizer provided 764 texts for task A and 178 texts for task B as a test set. The dataset was collected from four different domains including the SemEval-2010 Task 8 dataset, adverse drug effect (ADE) dataset, BBC news article dataset, and an inhouse data from the educational domain (educational app review).

To evaluate the performance of participants' systems, FIRE-2020 CEREX task organizers employed different strategies for task A and B. Standard evaluation measures, including precision, recall, and F1-score were applied to estimate the performance of participants' systems.

Table 4
Our result with other ranked teams.

Method	Precision	Recall	F1-Score
Results on Task A			
CSECU-DSG	**0.51**	**0.91**	**0.65**
SSN_NLP	0.46	0.87	0.60
Results on Task B			
SSN_NLP	**0.36**	**0.57**	**0.44**
CSECU-DSG	0.32	0.51	0.39

4.2. Parameter Settings

In the following, we describe the set of parameters that we have used to design our proposed model. We performed stratified k-Fold cross-validation with k=5 on the training set to select the best parameter setting. We used a pretrained sentence embedding model InferSent [13] to extract a 4096-dimensional embedding features in task A, whereas a 300-dimensional fastText [14] embedding model pretrained on Wikipedia with skip-gram is employed to extract the embedding features of each word in task B. As a classifier, we used the Scikit-learn implementation of the SVM classifier with the linear kernel (LinearSVC) and we determine the optimal value through cross-validation. We set the regularization parameter, C =10.0, random_state = 10.0 to control the pseudo-random number generation for shuffling the data, and the maximum number of iterations was set to 10000. We used the default settings for the rest of the parameters. In this paper, we reported the results based on these settings.

4.3. Results and Analysis

We used the full training dataset to train our proposed model and compare the performance based on test data against the participants' systems. The comparative results are presented in Table 4.

Results showed that our proposed system CSECU-DSG outperforms the other participant's system SSN_NLP for task A which deduces the effectiveness of our system for cause-effect relation identification from texts. However, our system lacks from SSN_NLP for task B. In task B, we employed the pretrained fastText model to extract the embedding features instead of the pretrained sentence embedding model. Since the sentence embedding model performs better for task A, it may works better for task B too. Besides, we employed a rich set of hand-crafted features ≈30 features, but didn't apply any effective feature selection and analysis techniques to remove the irrelevant and noisy features. Therefore, lacking of these settings hampered the performance of our model.

In order to estimate the contribution of our rule-based classifier (RBC) and different types of features in our proposed model, we performed the ablation study. In this regard, we removed RBC and different feature types at each time and repeated the experiment. Since the training set is imbalanced across classes, we took the 500 causal and 500 non-causal texts for effective ablation analysis. The results of our ablation study are reported in Table 5. It shows that when removing rule-based classifier (RBC) the results decreases 4% in terms of F1 score which deduced the contribution of RBC in our model. With this setting, we also analyze the effect of ablating different feature types. We observed a large decrease in performance while removing (InferSent+ngrams) features. A moderate decrease in performance is observed when removing InferSent and (InferSent+HCF) features. This observation deduced the importance of these features in our model.

Table 5
Feature ablation study of our proposed method for Task A based on training dataset.

Model	Precision	Recall	F1-Score	Accuracy
CSECU-DSG	0.71	0.87	0.78	0.76
Ablation of RBC from CSECU-DSG				
–CSECU-DSG without RBC	0.74	0.75	0.74	0.73
Feature ablation study using CSECU-DSG without RBC				
–InferSent	0.69	0.64	0.67	0.68
–ngrams	0.73	0.75	0.73	0.73
–HCF+ngrams	0.71	0.74	0.73	0.72
–InferSent+ngrams	0.45	0.44	0.45	0.45
–InferSent+HCF	0.63	0.63	0.63	0.62

5. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have presented our approach to tackle the problem defined in the FIRE-2020 CEREX task. We employed a feature fusion technique that incorporates the transfer learning embeddings and hand-crafted features in a unified feature vector. The generated features are then used to train an SVM based supervised model. Besides we employed a naive rule-based classifier on the top-of supervised model that accelerates the classification/annotation process. Experimental results demonstrated the efficacy of our feature combination which helped us to obtain the best result in task A and competitive performance in task B.

In the future, we have a plan to incorporate state-of-the-art deep learning techniques and explore effective hand-crafted features through feature selection. We also intend to extend our analysis on other causation patterns for exploring other possible uses of automatic extraction of causal-event.

References

[1] T. Dasgupta, R. Saha, L. Dey, A. Naskar, Automatic extraction of causal relations from text using linguistically informed deep neural networks, in: Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, 2018, pp. 306–316.

[2] R. Girju, D. I. Moldovan, et al., Text mining for causal relations., in: FLAIRS conference, 2002, pp. 360–364.

[3] R. Girju, Automatic detection of causal relations for question answering, in: Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering, 2003, pp. 76–83.

[4] B. Rink, C. A. Bejan, S. M. Harabagiu, Learning textual graph patterns to detect causal event relations., in: FLAIRS Conference, 2010.

[5] C. Hidey, K. McKeown, Identifying causal relations using parallel wikipedia articles, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1424–1433.

[6] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. K. Joshi, B. L. Webber, The penn discourse treebank 2.0., in: LREC, Citeseer, 2008.

[7] A. N. Chy, M. Z. Ullah, M. Aono, Microblog retrieval using ensemble of feature sets through

supervised feature selection, IEICE TRANSACTIONS on Information and Systems 100 (2017) 793–806.

[8] P. Chikersal, S. Poria, E. Cambria, Sentu: Sentiment analysis of tweets by combining a rule-based classifier with supervised learning, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 647–651.

[9] A. Bhardwaj, Y. Narayan, Vanraj, Pawan, M. Dutta, Sentiment analysis for indian stock market prediction using sensex and nifty, Procedia Computer Science 70 (2015) 85 – 91.

[10] A. N. Chy, M. Z. Ullah, M. Shajalal, M. Aono, Kdetm at ntcir-12 temporalia task: Combining a rule-based classifier with weakly supervised learning for temporal intent disambiguation., in: Proceedings of the 12th NTCIR ((NII Testbeds and Community for Information access Research) Conference, 2016.

[11] U. A. Siddiqua, T. Ahsan, A. N. Chy, Combining a rule-based classifier with ensemble of feature sets and machine learning techniques for sentiment analysis on microblog, in: 2016 19th International Conference on Computer and Information Technology (ICCIT), IEEE, 2016, pp. 304–309.

[12] C. Khoo, S. Chan, Y. Niu, The many facets of the cause-effect relation, in: The Semantics of Relationships, Springer, 2002, pp. 51–70.

[13] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, arXiv preprint arXiv:1705.02364 (2017).

[14] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146.

[15] P. Mirza, Extracting temporal and causal relations between events, in: Proceedings of the ACL 2014 Student Research Workshop, 2014, pp. 10–17.

[16] C. S. Khoo, Automatic identification of causal relations in text and their use for improving precision in information retrieval, Ph.D. thesis, 1995.

[17] Q. Gao, S. Yang, J. Chai, L. Vanderwende, What action causes this? towards naive physical action-effect prediction, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 934–945.

[18] X. Zhong, A. Sun, E. Cambria, Time expression analysis and recognition using syntactic token types and general heuristic rules, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 420–429.

[19] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A python natural language processing toolkit for many human languages, arXiv preprint arXiv:2003.07082 (2020).

[20] S. Loria, textblob documentation, Release 0.15 2 (2018).

[21] M. Honnibal, I. Montani, spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing, To appear 7 (2017).

[22] E. Loper, S. 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, 2002, pp. 63–70.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[24] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, V. Vapnik, Predicting time series with support vector machines, in: International Conference on Artificial Neural Networks, Springer, 1997, pp. 999–1004.