Exploring Word Representations on Time Expression Recognition

Sanxing Chen[*]
China University of Geosciences
Beijing 100083, China
sc3hn@virginia.edu

Guoxin Wang, Börje Karlsson
Microsoft Research
Beijing 100080, China
{guow,borje.karlsson}@microsoft.com

Abstract

Time expression extraction has attracted longstanding interest over time, due to its great importance in many downstream tasks of Natural Language Processing (NLP) and Information Retrieval (IR). Although current approaches, either rule-based or learning-based, can achieve impressive performance in major datasets, they usually rely heavily on handcrafted rules or task-specific pre-tagging features. Recent advances in pretrained word representations motivate us to explore semi-supervised approaches for this task. We first show that simple neural architectures built on top of pre-trained word representations perform competitively and efficiently on time expression recognition. Then we further explore several design choices focusing on the need of contextualization and the training resource requirements for this type of time expression taggers.

1 Introduction

Time expressions play an important role in human languages. Many time-aware NLP tasks such as question answering (Llorens et al., 2015), textual entailment (Wang and Zhang, 2008), summarization (Aramaki et al., 2009), usually require the ability to deal with time information in text.

As a long established field, researchers have long discovered that time expressions are usually formed by a limited number of words in a loose structure. Thus many rule-based systems have been proved to be able to solve a large part of this task. Unfortunately, those rule-based approaches are neither computational efficient by design nor capable to deal with some vague or ambiguous cases. Even though some learning-based or hybrid systems are proposed to solve these problems by leveraging statistic information from the training corpus, they still heavily rely on some hand-engineered patterns for deterministic matching or token types pre-tagging, which makes it difficult to scale to other domains or languages. Moreover, they are not intelligent enough to recognize time expressions in difficult contexts due to the limited information provided by syntactic or lexical features. For example, the word *Fall* has two absolute different meanings in the phrases "Siege and *Fall* of Port Arthur" and "Spring and *Fall* of Port Arthur".[1] Specifically, the later one should be recognize as a time expression, but the two meanings cannot be disambiguated by either syntactic features or un-contextualized word sense.

In this paper, we explore neural time expression tagging models based on two kinds of pretrained word representations. Our work is motivated by two intuitions. First, since word representations are pre-trained on large corpora, they can generate informative representations without being influenced by the small corpus size, thus the model doesn't require any task-specific features and might be more reliable under low-resource conditions. Second, contextual word representations are designed to capture different word senses depending on the context in which they are located, so they could distinguish time expressions from complex contexts.

We summarize the our contributions as follows:

- We adopt neural network models using word representations in time expression recognition. Experiments on simple neural architectures with contextual word representations show state-of-the-art performance on standard benchmarks.

- We further conduct experiments to probe the

[*] Work was done during an internship at Microsoft Research Asia.

Technical report

[1] This example is cited from the WikiWars dataset (Mazur and Dale, 2010)

need for contextual information in neural time expression taggers. Our empirical observations confirm our intuitions and offer insight for future model design.

2 Related Work

Since TempEval-2 (Pustejovsky et al., 2009) initially introduce a shared task of determining the extent of the time expressions in text, a series of TempEval shared tasks (UzZaman et al., 2013; Bethard et al., 2015, 2016, 2017) have been launched and attracted great research interest.

2.1 Rule-based Approaches

Rule-based approaches usually use deterministic rules for matching. SUTIME (Chang and Manning, 2012) designs three types of rules, *i.e.*, text regexes, compositional rules and filtering rules to form a 3-layered temporal pattern language. In addition to serving as a critical component in the Stanford CoreNLP library (Manning et al., 2014), its text regexes for time token matching are adopted by lots of later proposed methods (Zhong and Cambria, 2018; Ding et al., 2019). Another rule-based system, Syntime (Zhong et al., 2017), designs heuristic rules on top of a fine-grained token type system. Other sophisticated rule-based systems are also widely available, such as the Recognizers-Text [2] which supports time expression recognition in a multi-lingual setting.

2.2 Learning-based Approaches

Because of the small size of commonly used corpus, learning-based approaches often require the use of hand-crafted features to exploit human insights into the problem.

Zhong and Cambria (2018) use a conditional random field model trained to model time expression under a task-specific constituent-based tagging scheme named TOMN. The model is fed by carefully designed features, *i.e.*, regular expression pre-tags and lemma features. Ding et al. (2019) use an Extended Budgeted Maximum Coverage model to learn to select patterns which are automatically generated from training text. However, they still need to manually design a fine-grained token type system to pre-tag input text. In addition, their method cannot utilizes the contextual information for disambiguation, resulting in a gap between their model performance and other approaches on the TempEval-3 dataset.

2.3 Neural Network Approaches

Recent advances in deep learning algorithms reveal that neural networks can learn good representations from input distributions (Bengio et al., 2003; Mikolov et al., 2013). But only limited effort have been put into neural approaches for this task.

Olex et al. (2018) use neural networks as a small component to disambiguate two entity types “Period” and “Calendar-Interval” with contextual information in their temporal expression normalization system. Etcheverry and Wonsever (2017) conduct experiments on mainly Spanish time expression recognition by using neural networks and word embeddings. They show that distributed word representations can capture time information to some extent. They also conduct experiments to explore the architectures choosing on this task. The performance of their proposed models is inferior to a rule-based system HeidelTime (Strötgen and Gertz, 2010) by a large margin. Note that with the progress of research in this field, many current state-of-the-art systems have now surpassed HeidelTime a lot.

3 Models

In this section, we provide a brief description of BERT and present the models used in our experiments.

BERT (Devlin et al., 2018) stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. In contrast to some previous proposed language model pre-training approaches like OpenAI GPT (Radford et al., 2018), BERT explicitly models the context from both directions, which is arguably important to token-level tasks such as named entity recognition (NER) (Sang and De Meulder, 2003) and SQuAD question answering (Rajpurkar et al., 2016).

3.1 BERT-based Tagging Models

We use the BERT-Base model (cased, 12-layer, 768-hidden, 12-heads, 110M parameters) with a single layer linear classifier on top of it. Following the advice for token tagging task in the original paper, we first tokenize the input sentence using WordPiece tokenizer and feed the hidden state of the first sub-token to the classifier. The clas-

[2] https://github.com/microsoft/Recognizers-Text

Datasets	Timex Systems	Strict Match			Relaxed Match		
		Pr.	*Re.*	F_1	*Pr.*	*Re.*	F_1
TE-3	SynTime	91.43	**92.75**	**92.09**	94.29	**95.65**	**94.96**
	TOMN	**92.59**	90.58	91.58	95.56	93.48	94.51
	PTime	85.19	83.33	84.25	92.59	90.58	91.58
	BERT (Fine-tune)	92.48	89.13	90.77	**96.24**	92.75	94.46
WikiWars	SynTime	80.00	80.22	80.11	92.16	92.41	92.29
	TOMN	84.57	80.48	82.47	96.23	92.35	94.25
	PTime	86.86	87.57	87.21	95.98	96.76	96.37
	BERT (Fine-tune)	**95.46**	**96.60**	**96.03**	**98.24**	**99.41**	**98.82**
Tweets	SynTime	89.52	94.07	91.74	93.55	98.31	95.87
	TOMN	90.69	94.51	92.56	93.52	97.47	95.45
	PTime	**92.92**	94.09	93.50	**97.92**	**99.16**	**98.53**
	BERT (Fine-tune)	92.21	**94.94**	**93.56**	95.08	97.89	96.47

Table 1: Performance of our proposed methods compared with state-of-the-art systems on three benchmark datasets. Statistic of previous systems are cited from their original papers. The best-performing system is bolded.

sification layer outputs a type prediction for each input token under the standard B(egin), I(nside), and O(utside) labeling scheme. We fine-tune the entire model on the training corpus to predict token types.

4 Experiments

Evaluation Metric. We report precision, recall and F1 measure in both strict match and relaxed match. Results are averaged over 5 random seeds.
Datasets. We evaluate our model on a collection of three commonly used datasets, *i.e.*, TimeBank (Pustejovsky et al., 2003), the WikiWars (Mazur and Dale, 2010) and the Tweets (Zhong et al., 2017). TimeBank and WikiWars are both formal text corpus, while Tweets consists of informal texts crawled from the web. Texts in TimeBank lie in news domain, while WikiWars contains 22 documents from English Wikipedia that describe the historical course of wars.

Dataset	Docs	Words	Timex
TimeBank (train)	183	61,418	1,243
TempEval-3 (test)	20	6,375	138
WikiWars (train)	17	98,746	2,228
WikiWars (test)	5	19,052	363
Tweets (all)	942	18,199	1,127

Table 2: Dataset statistics

For the purpose of fair comparisons, we strictly follow the same data splitting strategy used in previous works (Zhong and Cambria, 2018; Ding et al., 2019). For TimeBank, we use TimeBank 1.2 corpus[3] for training and TempEval-3 Platinum (UzZaman et al., 2013) dataset for testing. The statistics of all these datasets are listed in Table 2.
Comparison Systems. We list three baseline systems (*i.e.*, SynTime, TOMN and PTime) which cover all the state-of-the-art results on the three datasets mentioned before.
Implementation Details. According to BERT's advice on doing sequence tagging task, we use a dropout probability of 0.1 on the representation output layer. We use a batch size of 16 for 8 epochs and Adam (lr=5e-5) for model optimization. Hyper-parameters are chosen by grid searching on a development set.
Benchmark Results. The results show that the BERT-based model is comparable to state-of-the-art systems on both TempEval-3 and Tweets datasets. On Wikiwars, our model outperforms the previous state-of-the-art by a large margin, which establishes a new state-of-the-art result. It improves the strict match F_1 by 8.82%, and the relaxed match F_1 by 2.45%.

5 Understanding Representations

Even if we know fine-tuning BERT can achieve great results, we're still unclear about how word representations benefit our model and what's the essential neural architecture which is sufficient for this task. So we further conduct several probing experiments by using BERT as a pure feature extractor (that means we freeze all param-

[3] See corpus LDC2006T08 in the LDC catalogue

	Probing Models	Strict F_1	Relaxed F_1
GloVe	Linear	16.90±.30	78.51±.21
	MLP	16.14±.62	81.52±.23
	LSTM	43.36±.90	84.68±.94
	BiLSTM+MLP	83.85±.44	93.14±.22
BERT	Linear	75.24±.45	88.50±.15
	MLP	87.18±.85	94.95±.35
	LSTM	91.10±.42	95.62±.20
	BiLSTM+MLP	93.24±.30	96.54±.34
BERT	Fine-tune	96.03±.35	98.82±.26

Table 3: Results of probing experiments on WikiWars.

eters of BERT during training) and additionally use GloVe (Pennington et al., 2014) as an representative uncontextualized word representations to compared with.

We feed the features extracted from these two representations into three different neural models, i.e., a linear model, a LSTM (single layer with 200 hidden units) model, a multilayer perceptron (MLP: a single 1000d hidden layer activated by ReLU) and a full-featured model with both BiLSTM layers and MLP. The LSTM model can be seen as a task-specific contextualizer while the MLP model can be seen as additional parameters and nonlinearity. The two models have nearly the same number of parameters.[4]

We choose to perform this suite of experiments on the WikiWars dataset because TempEval-3 has significant fewer testing data which leads to instability in testing and using the Tweets dataset introduces domain inconsistency.

5.1 Results and Discussion

Table 3 presents the performance of BERT-based and GloVe-based probing models. In GloVe-based models, there is always a big gap between strict and relaxed matching scores. The relaxed matching score can be seen as a measurement of the model's capability to find time expressions from texts, while the strict matching measurement requires the model to correctly distinguish the boundary of time expressions. Actually, most time expressions contain time tokens which can be recognized by itself, so both simple rule-based systems and non-contextual word representation can do relax matching. But strict matching cares about the boundary which is usually composed by general modifier tokens and numeral tokens in a loose structure (Zhong and Cambria, 2018), so it requires more contextual information.

From the results of BERT-based models, We find that the additional parameters from LSTM and MLP models can significantly improve the linear baseline. It reveals that the model needs a minimal structure to learn task-specific knowledge. Given that the results from GloVe have already proved the necessity of contextual information in this task, comparisons between LSTM and MLP models suggest that BERT-based models rely less on additional contextual information.

The results also show that the performance of the full-featured model is comparable to fine-tuned BERT model. We suspect adding more parameters of the network would be sufficient to fill the small gap.

5.2 Resource Experiment

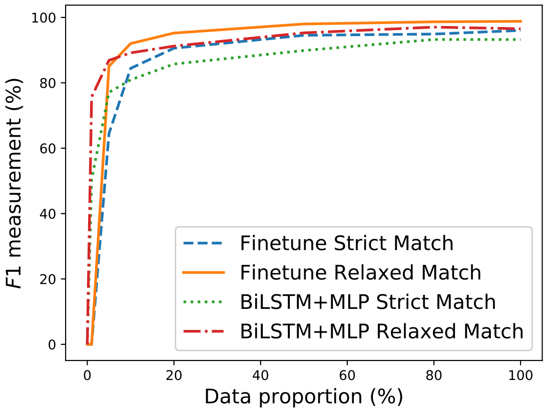

Figure 1: Performance of BERT (Fine-tune) and BERT-BiLSTM-MLP in a simulated resource-poor setup.

In order to figure out how much training resource do neural models really need in this task, we simulate a low-resource scenario in which we gradually increase the amount of training resource. We also conduct this experiment on the WikiWars dataset. The results in Figure 1 show that BERT-based taggers can actually learn very well even if very little data are provided.

6 Conclusion

We study word representations based neural models on time expression recognition. The results of our experiments show that neural approaches are competitive in this task, even under low-resource conditions. In addition, fine-tuning a contextualizer BERT can establish a new state-of-the-art in

[4] We refer to the experiments setup in Liu et al. (2019).

one commonly used dataset. We further probe the need of contextual information in this task, the results confirm our intuitions.

References

Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Mashuichi, and Kazuhiko Ohe. 2009. Text2table: Medical text summarization system based on named entity recognition and modality identification. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 185–192. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. journal of machine learning research, vol. 3, no.

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. Semeval-2015 task 6: Clinical tempeval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. Semeval-2016 task 12: Clinical tempeval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062.

Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. Semeval-2017 task 12: Clinical tempeval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572.

Angel X Chang and Christopher Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Wentao Ding, Guanji Gao, Linfeng Shi, and Yuzhong Qu. 2019. A Pattern-based Approach to Recognizing Time Expressions. In *Thirty-Third AAAI Conference on Artificial Intelligence*.

Mathias Etcheverry and Dina Wonsever. 2017. Time expressions recognition with word vectors and neural networks. In *24th International Symposium on Temporal Representation and Reasoning (TIME 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. 2019. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015. Semeval-2015 task 5: Qa tempeval-evaluating temporal information understanding with question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 792–800.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Pawet Mazur and Robert Dale. 2010. Wikiwars: A new corpus for research on temporal expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 913–922. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Amy Olex, Luke Maffey, Nicholas Morgan, and Bridget McInnes. 2018. Chrono at semeval-2018 task 6: A system for normalizing temporal expressions. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 97–101.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

James Pustejovsky, Marc Verhagen, Xue Nianwen, Robert Gaizauskas, Mark Hepple, Frank Schilder, Graham Katz, Roser Saurí, Estela Saquete, Tommaso Caselli, et al. 2009. Tempeval2: Evaluating events, time expressions and temporal relations. *SemEval Task Proposal*.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Jannik Strötgen and Michael Gertz. 2010. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324. Association for Computational Linguistics.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 1–9.

Rui Wang and Yajing Zhang. 2008. Recognizing textual entailment with temporal expressions in natural language texts. In *2008 IEEE International Workshop on Semantic Computing and Applications*, pages 109–116. IEEE.

Xiaoshi Zhong and Erik Cambria. 2018. Time expression recognition using a constituent-based tagging scheme. In *WWW*, pages 983–992.

Xiaoshi Zhong, Aixin Sun, and Erik Cambria. 2017. 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)*, pages 420–429.