Model-Portability Experiments for Textual Temporal Analysis

Oleksandr Kolomiyets, Steven Bethard and Marie-Francine Moens

Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A, Heverlee, 3001, Belgium
{oleksandr.kolomiyets, steven.bethard, sien.moens}@cs.kuleuven.be

Abstract

We explore a semi-supervised approach for improving the portability of time expression recognition to non-newswire domains: we generate additional training examples by substituting temporal expression words with potential synonyms. We explore using synonyms both from WordNet and from the Latent Words Language Model (LWLM), which predicts synonyms in context using an unsupervised approach. We evaluate a state-of-the-art time expression recognition system trained both with and without the additional training examples using data from TempEval 2010, Reuters and Wikipedia. We find that the LWLM provides substantial improvements on the Reuters corpus, and smaller improvements on the Wikipedia corpus. We find that WordNet alone never improves performance, though intersecting the examples from the LWLM and WordNet provides more stable results for Wikipedia.

1 Introduction

The recognition of time expressions such as *April 2011*, *mid-September* and *early next week* is a crucial first step for applications like question answering that must be able to handle temporally anchored queries. This need has inspired a variety of shared tasks for identifying time expressions, including the Message Understanding Conference named entity task (Grishman and Sundheim, 1996), the Automatic Content Extraction time normalization task (http://fofoca.mitre.org/tern.html) and the TempEval 2010 time expression task (Verhagen et al., 2010). Many researchers competed in these tasks, applying both rule-based and machine-learning approaches (Mani and Wilson, 2000; Negri and Marseglia, 2004; Hacioglu et al., 2005; Ahn et al., 2007; Poveda et al., 2007; Strötgen and Gertz 2010; Llorens et al., 2010), and achieving F1 measures as high as 0.86 for recognizing temporal expressions.

Yet in most of these recent evaluations, models are both trained and evaluated on text from the same domain, typically newswire. Thus we know little about how well time expression recognition systems generalize to other sorts of text. We therefore take a state-of-the-art time recognizer and evaluate it both on TempEval 2010 and on two new test sets drawn from Reuters and Wikipedia.

At the same time, we are interested in helping the model recognize more types of time expressions than are available explicitly in the newswire training data. We therefore introduce a semi-supervised approach for expanding the training data, where we take words from temporal expressions in the data, substitute these words with likely synonyms, and add the generated examples to the training set. We select synonyms both via WordNet, and via predictions from the Latent Words Language Model (LWLM) (Deschacht and Moens, 2009). We then evaluate the semi-supervised model on the TempEval, Reuters and Wikipedia test sets and observe how well the model has expanded its temporal vocabulary.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 271–276,
Portland, Oregon, June 19-24, 2011.

2 Related Work

Semi-supervised approaches have been applied to a wide variety of natural language processing tasks, including word sense disambiguation (Yarowsky, 1995), named entity recognition (Collins and Singer, 1999), and document classification (Surdeanu et al., 2006).

The most relevant research to our work here is that of (Poveda et al., 2009), which investigated a semi-supervised approach to time expression recognition. They begin by selecting 100 time expressions as seeds, selecting only expressions that are almost always annotated as times in the training half of the Automatic Content Extraction corpus. Then they begin an iterative process where they search an unlabeled corpus for patterns given their seeds (with patterns consisting of surrounding tokens, parts-of-speech, syntactic chunks etc.) and then search for new seeds given their patterns. The patterns resulting from this iterative process achieve F1 scores of up to 0.604 on the test half of the Automatic Content Extraction corpus.

Our approach is quite different from that of (Poveda et al., 2009) – we use our training corpus for learning a supervised model rather than for selecting high precision seeds, we generate additional training examples using synonyms rather than bootstrapping based on patterns, and we evaluate on Reuters and Wikipedia data that differ from the domain on which our model was trained.

3 Method

The proposed method implements a supervised machine learning approach that classifies each chunk-phrase candidate top-down starting at the parse tree root provided by the OpenNLP parser. Time expressions are identified as phrasal chunks with spans derived from the parse as described in (Kolomiyets and Moens, 2010).

3.1 Basic TempEval Model

We implemented a logistic regression model with the following features for each phrase-candidate:

- The head word of the phrase
- The part-of-speech tag of the head word
- All tokens and part-of-speech tags in the phrase as a bag of words
- The word-shape representation of the head word and the entire phrase, e.g. `Xxxxx 99` for the expression *April 30*
- The condensed word-shape representation for the head word and the entire phrase, e.g. `X(x) (9)` for the expression *April 30*
- The concatenated string of the syntactic types of the children of the phrase in the parse tree
- The depth in the parse tree

3.2 Lexical Resources for Bootstrapping

Sparsity of annotated corpora is the biggest challenge for any supervised machine learning technique and especially for porting the trained models onto other domains. To overcome this problem we hypothesize that knowledge of semantically similar words, like temporal triggers, could be found by associating words that do not occur in the training set to similar words that do occur in the training set. Furthermore, we would like to learn these similarities automatically to be independent of knowledge sources that might not be available for all languages or domains. The first option is to use the Latent Words Language Model (LWLM) (Deschacht and Moens, 2009) – a language model that learns from an unlabeled corpus how to provide a weighted set of synonyms for words in context. The LWLM model is trained on the Reuters news article corpus of 80 million words.

WordNet (Miller, 1995) is another resource for synonyms widely used in research and applications of natural language processing. Synonyms from WordNet seem to be very useful for bootstrapping as they provide replacement words to a specific word in a particular sense. For each synset in WordNet there is a collection of other "sister" synsets, called coordinate terms, which are topologically located under the same hypernym.

3.3 Bootstrapping Strategies

Having a list of synonyms for each token in the sentence, we can replace one of the original tokens by its synonym while still mostly preserving the sentence semantics. We choose to replace just the headword, under the assumption that since temporal trigger words usually occur at the headword position, adding alternative synonyms for the headword should allow our model to learn temporal triggers that did not appear in the training data.

		Basic TempEval Model	Bootstrapped Models			
			LWLM	WordNet 1st Sense	WordNet Pseudo-Lesk	LWLM+ WordNet
TempEval 2010	# *Syn*	0	1	1	1	2
	P	0.916	0.865	0.881	0.894	0.857
	R	0.770	0.807	0.773	0.781	0.830
	F1	0.834	0.835	0.824	0.833	0.829
Reuters	# *Syn*	0	5	7	6	4
	P	0.896	0.841	0.820	0.839	0.860
	R	0.679	0.812	0.721	0.717	0.742
	F1	0.773	**0.826**	0.767	0.773	**0.796**
Wikipedia	# *Syn*	0	3	1	6	5
	P	0.959	0.924	0.922	0.909	0.913
	R	0.770	0.830	0.781	0.820	0.844
	F1	0.859	**0.874**	0.858	0.862	**0.877**

Table 1: Precision, recall and F1 scores for all models on the source (TempEval 2010) and target (Reuters and Wikipedia) domains. Bootstrapped models were asked to generate between one and ten additional training examples per instance. The maximum P, R, F1 and the number of synonyms at which this maximum was achieved are given in the P, R, F1 and # Syn rows. F1 scores more than 0.010 above the Basic TempEval Model are marked in bold.

We designed the following bootstrapping strategies for generating new temporal expressions:

- **LWLM**: the phrasal head is replaced by one of the LWLM synonyms.
- **WordNet 1st Sense**: Synonyms and coordinate terms for the most common sense of the phrasal head are selected and used for generating new examples of time expressions.
- **WordNet Pseudo-Lesk**: The synset for the phrasal head is selected as having the largest intersection between the synset's words and the LWLM synonyms. Then, synonyms and coordinate terms are used for generating new examples of time expressions.
- **LWLM+WordNet**: The intersection of the LWLM synonyms and the WordNet synset found by pseudo-Lesk are used.

In this way for every annotated time expression we generate n new examples ($n \in [1,10]$) and use them for training bootstrapped classification models.

4 Experimental Setup

The tested model is trained on the official TempEval 2010 training data with 53450 tokens and 2117 annotated TIMEX3 tokens. For testing the portability of the model to other domains we annotated two small target domain document collections with TIMEX3 tags. The first corpus is 12 Reuters news articles from the Reuters corpus (Lewis et al., 2004), containing 2960 total tokens and 240 annotated TIMEX3 tokens (inter-annotator agreement 0.909 F1-score). The second corpus is the Wikipedia article for Barak Obama (http://en.wikipedia.org/wiki/Obama), containing 7029 total tokens and 512 annotated TIMEX3 tokens (inter-annotator agreement 0.901 F1-score).

The basic TempEval model is evaluated on the source domain (TempEval 2010 evaluation set – 9599 tokens in total and 269 TIMEX3 annotated tokens) and target domain data (Reuters and Wikipedia) using the TempEval 2010 evaluation metrics. Since porting the model onto other domains usually causes a performance drop, our experiments are focused on improving the results by employing different bootstrapping strategies[1].

5 Results

The recognition performance of the model is reported in Table 1 (column "Basic TempEval Model") for the source and the target domains. The basic TempEval model itself achieves F1-score of 0.834 on the official TempEval 2010 evaluation corpus and has a potential rank 8 among 15 participated systems. The top seven TempEval-2 systems achieved F1-score between 0.83 and 0.86.

[1] The annotated datasets are available at http://www.cs.kuleuven.be/groups/liir/software.php

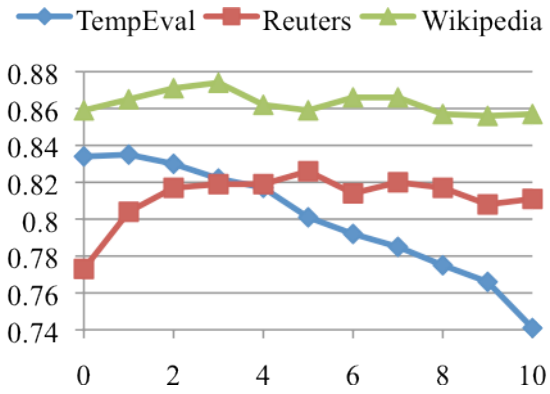

Figure 1: F1 score of the LWLM bootstrapping strategy, generating from zero to ten additional training examples per instance.

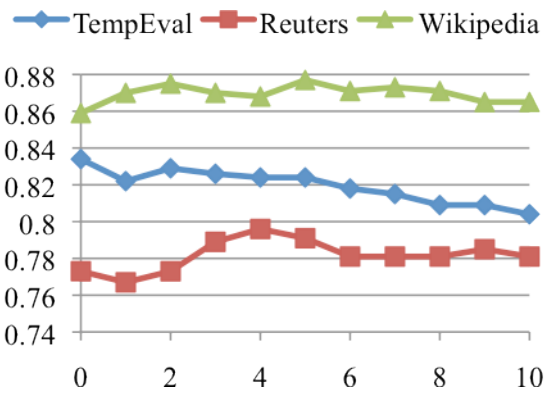

Figure 2: F1 score of the LWLM+WordNet bootstrapping strategy, generating from zero to ten additional training examples per instance.

However, this model does not port well to the Reuters corpus (0.773 vs. 0.834 F1-score). For the Wikipedia-based corpus, the basic TempEval model actually performs a little better than on the source domain (0.859 vs. 0.834 F1-score).

Four bootstrapping strategies were proposed and evaluated. Table 1 shows the maximum F1 score achieved by each of these strategies, along with the number of generated synonyms (between one and ten) at which this maximum was achieved. None of the bootstrapped models outperformed the basic TempEval model on the TempEval 2010 evaluation data, and the WordNet 1st Sense strategy and the WordNet Pseudo-Lesk strategy never outperformed the basic TempEval model on any corpus.

However, for the Reuters and Wikipedia corpora, the LWLM and LWLM+WordNet bootstrapping strategies outperformed the basic TempEval model. The LWLM strategy gives a large boost to model performance on the Reuters corpus from 0.773 up to 0.826 (a 23.3% error reduction) when using the first 5 synonyms. This puts performance on Reuters near performance on the TempEval domain from which the model was trained (0.834). This suggests that the (Reuters-trained) LWLM is finding exactly the right kinds of synonyms: those that were not originally present in the TempEval data but are present in the Reuters test data. On the Wikipedia corpus, the LWLM bootstrapping strategy results in a moderate boost, from 0.859 up to 0.874 (a 10.6% error reduction) when using the first three synonyms. Figure 1 shows that using more synonyms with this strategy drops performance on the Wikipedia corpus back down to the level of the basic TempEval model.

The LWLM+WordNet strategy gives a moderate boost on the Reuters corpus from 0.773 up to 0.796 (a 10.1% error reduction) when four synonyms are used. Figure 2 shows that using six or more synonyms drops this performance back to just above the basic TempEval model. On the Wikipedia corpus, the LWLM+WordNet strategy results in a moderate boost, from 0.859 up to 0.877 (a 12.8% error reduction), with five synonyms. Using additional synonyms results in a small decline in performance, though even with ten synonyms, the performance is better than the basic TempEval model.

In general, the LWLM strategy gives the best performance, while the LWLM+WordNet strategy is less sensitive to the exact number of synonyms used when expanding the training data.

6 TempEval Error Analysis

We were curious why synonym-based bootstrapping did not improve performance on the source-domain TempEval 2010 data. An error analysis suggested that some time expressions might have been left unannotated by the human annotators. Two of the authors re-annotated the TempEval evaluation data, finding inter-annotator agreement of 0.912 F1-score with each other, but only 0.868 and 0.887 F1-score with the TempEval annotators, primarily due to unannotated time expressions such as *23-year*, *a few days* and *third-quarter*.

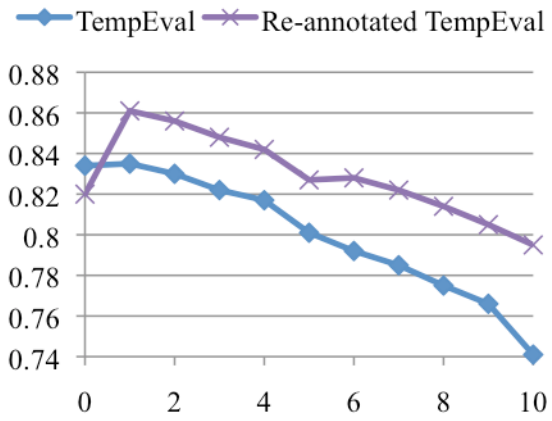

Figure 3: F1 score of the LWLM bootstrapping strategy, comparing performance on the original TempEval data to the re-annotated version.

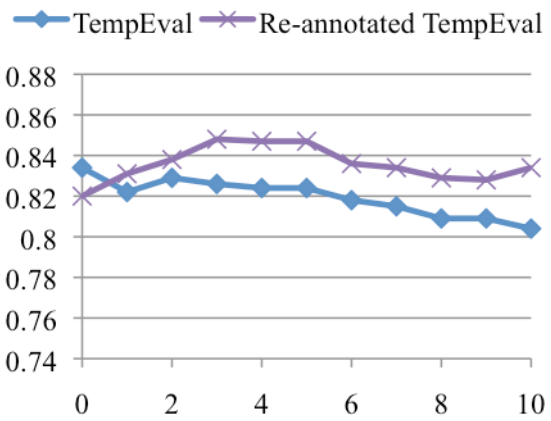

Figure 4: F1 score of the LWLM+WordNet bootstrapping strategy, comparing performance on the original TempEval data to the re-annotated version.

Using this re-annotated TempEval 2010 data[2], we re-evaluated the proposed bootstrapping techniques. Figure 3 and Figure 4 compare performance on the original TempEval data to performance on the re-annotated version. We now see the same trends for the TempEval data as were observed for the Reuters and Wikipedia corpora: using a small number of synonyms from the LWLM to generate new training examples leads to performance gains. The LWLM bootstrapping model using the first synonym achieves 0.861 F1 score, a 22.8% error reduction over the baseline of 0.820 F1 score.

7 Discussion and Conclusions

We have presented model-portability experiments on time expression recognition with a number of bootstrapping strategies. These bootstrapping strategies generate additional training examples by substituting temporal expression words with potential synonyms from two sources: WordNet and the Latent Word Language Model (LWLM).

Bootstrapping with LWLM synonyms provides a large boost for Reuters data and TempEval data and a decent boost for Wikipedia data when the top few synonyms are used. Additional synonyms do not help, probably because they are too newswire-specific: both the contexts from the TempEval training data and the synonyms from the Reuters-trained LWLM come from newswire text, so the lower synonyms are probably more domain-specific.

Intersecting the synonyms generated by the LWLM and by WordNet moderates the LWLM, making the bootstrapping strategy less sensitive to the exact number of synonyms used. However, while the intersected model performs as well as the LWLM model on Wikipedia, the gains over the non-bootstrapped model on Reuters and TempEval data are smaller.

Overall, our results show that when porting time expression recognition models to other domains, a performance drop can be avoided by synonym-based bootstrapping. Future work will focus on using synonym-based expansion in the contexts (not just the time expressions headwords), and on incorporating contextual information and syntactic transformations.

Acknowledgments

This work has been funded by the Flemish government as a part of the project AMASS++ (Advanced Multimedia Alignment and Structured Summarization) (Grant: IWT-SBO-060051).

References

David Ahn, Joris van Rantwijk, and Maarten de Rijke. 2007. A Cascaded Machine Learning Approach to Interpreting Temporal Expressions. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL-HLT 2007).

[2] Available at http://www.cs.kuleuven.be/groups/liir/software.php

Michael Collins and Yoram Singer. 1999. Unsupervised Models for Named Entity Classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 100–110, College Park, MD. ACL.

Koen Deschacht and Marie-Francine Moens. 2009. Using the Latent Words Language Model for Semi-Supervised Semantic Role Labeling. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: A Brief History. In *Proceedings of the 16th Conference on Computational Linguistics*, pp. 466–471.

Kadri Hacioglu, Ying Chen, and Benjamin Douglas 2005. Automatic Time Expression Labeling for English and Chinese Text. In Gelbukh, A. (ed.) *CICLing 2005*. LNCS, vol. 3406, pp. 548–559. Springer, Heidelberg.

Oleksandr Kolomiyets, Marie-Francine Moens. 2010. KUL: Recognition and Normalization of Temporal Expressions. In *Proceedings of SemEval-2 5th Workshop on Semantic Evaluation*. pp. 325-328. Uppsala, Sweden. ACL.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Machine Learning Research.* 5: 361-397

Inderjeet Mani, and George Wilson. 2000. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 69-76, Morristown, NJ. ACL.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11): 39-41.

Matteo Negri, and Luca Marseglia. 2004. Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004. Technical Report, ITC-irst, Trento.

Hector Llorens, Estela Saquete, and Borja Navarro. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval 2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 284–291, Uppsala, Sweden. ACL.

Jordi Poveda, Mihai Surdeanu, and Jordi Turmo. 2007. A Comparison of Statistical and Rule-Induction Learners for Automatic Tagging of Time Expressions in English. In *Proceedings of the International Symposium on Temporal Representation and Reasoning*, pp. 141-149.

Jordi Poveda, Mihai Surdeanu, and Jordi Turmo. 2009. An Analysis of Bootstrapping for the Recognition of Temporal Expressions. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pp. 49-57, Stroudsburg, PA, USA. ACL.

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*, pp. 321–324, Uppsala, Sweden. ACL.

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2006. A Hybrid Approach for the Acquisition of Information Extraction Patterns. In *Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining* (ATEM 2006). ACL.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval 2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 57–62, Uppsala, Sweden. ACL.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189–196, Cambridge, MA. ACL.