Identifying Temporality of Word Senses Based on Minimum Cuts

M. Hasanuzzaman,[1,3] **G. Dias,**[1,2] **S. Ferrari,**[1,2] **Y. Mathet,**[1,2] **and A. Way**[3]
[1]Université de Caen Normandie, Caen, France
[2]GREYC UMR 6072, Caen, France
[3]ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland

Abstract

The ability to capture time information is essential to many natural language processing and information retrieval applications. Therefore, a lexical resource associating word senses to their temporal orientation might be crucial for the computational tasks aiming at the interpretation of language of time in texts. In this paper, we propose a semi-supervised minimum cuts strategy that makes use of WordNet glosses and semantic relations to supplement WordNet entries with temporal information. Intrinsic and extrinsic evaluations show that our approach outperforms prior semi-supervised non-graph classifiers.

1 Introduction

Recognizing temporal information can significantly improve the functionality of information retrieval (Campos et al., 2014) and natural language processing (Mani et al., 2005) applications.

Most text applications have been relying on rule-based time taggers such as HeidelTime (Strötgen and Gertz, 2015) or SUTime (Chang and Manning, 2012) to identify and normalize time mentions in texts. Although interesting levels of performance have been seen (UzZaman et al., 2013), their coverage is limited to the finite number of rules they implement. Let's take the following sentence: "*Apple's iPhone is currently one of the most popular smartphone*". When labeled by SUTime[1] or HeidelTime[2], the adverb *currently* is correctly tagged with the PRESENT_REF value. However, if we change the sentence to "*Apple's iPhone is one of the most popular smartphones at the present day*", no temporal mention is found, although one may expect that within this context *currently* and *present day* share some equivalent temporal dimension. Such systems would certainly benefit from the existence of a temporal resource enumerating a large set of possible time variants (Kuzey et al., 2016).

In parallel, new trends have emerged in the context of human temporal orientation (Schwartz et al., 2015). The underlying idea is to understand how past, present, and future emphasis in text may affect people's finances, health, and happiness. For that purpose, temporal classifiers are built to detect the overall temporal dimension of a given sentence. For instance, the following Facebook post "*can't wait to get a pint tonight*" would be tagged as FUTURE. Successful features include timexes, specific temporal (past, present, future) words from a commercial dictionary, but also *n*-grams, thus indicating that temporality may be embodied by multi-word terms, whose temporal orientation is unknown.

As a consequence, discovering the temporal orientation of words is a challenging issue that may benefit many text applications. Whereas most prior studies have focused on temporal expressions and events, there has been a lack of work looking at the temporal orientation of word senses. In this paper, we focus on automatically time-tagging word senses in WordNet (Miller, 1995) as *past*, *present*, *future*, or *atemporal* based on their glosses and relational semantic structures in the line of Dias et al. (2014) and Hasanuzzaman et al. (2014b). In particular, we propose a semi-supervised graph-based strategy that relies on the max-flow min-cut theorem (Papadimitriou and Steiglitz, 1998; Blum and Chawla, 2001), that finds successive minimum cuts in a connected graph to time-tag each synset as one of the four

[1]http://nlp.stanford.edu:8080/sutime/process
[2]http://heideltime.ifi.uni-heidelberg.de/heideltime/

Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), pages 22–30,
Berlin, Germany, August 7-12, 2016.

dimensions. Compared to previous work based on propagation strategies (Dias et al., 2014; Hasanuzzaman et al., 2014), the exploration of WordNet's graph structure with minimum cuts allows us to independently model both temporal connotation and semantic denotation. In order to evaluate our proposal, both intrinsic (inter-annotator agreement and temporal sense classification) and extrinsic (temporal sentence classification and temporal relation annotation) evaluations have been performed. In both cases, the proposed methodology outperformed state-of-the-art approaches.

2 Related Work

Dias et al. (2014) developed TempoWordNet (TWnL), an extension of WordNet, where each synset is augmented with its temporal connotation (*past*, *present*, *future*, or *atemporal*). It mainly relies on the quantitative analysis of the glosses associated to synsets, and on the use of the resulting vector space model representations for semi-supervised synset classification. In particular, temporal classifiers are learned over manually labeled synsets (seed list), and new learning synsets are chosen based on their specific semantic relation (e.g. hyponymy) with synsets from the seed list. Their class is given by the synset they have been propagated from. This process is iterated until cross-validation accuracy drops. The final classifier is used to time-tag all WordNet synsets.

While Hasanuzzaman et al. (2014) show that TWnL can be useful to time-tag web queries, less comprehensive results are shown in Filannino and Nenadic (2014), where TWnL learning features do not lead to any classification improvements. Moreover, Dias et al. (2014) mention that exclusive semantic propagation is error-prone as some semantic relations do not preserve temporal connotation. As a consequence, Hasanuzzaman et al. (2014b) defined two different propagation strategies: probabilistic and hybrid, leading to TWnP and TWnH, respectively. They follow the exact same idea of Dias et al. (2014), but for probabilistic propagation, new synsets are chosen from the most confidently classified synsets over the whole of WordNet at each iteration. In addition, for the hybrid expansion, new learning instances are included if they are highly representative of a given class but at the same time demonstrate high average semantic similarity over the seed list. Although some slight improvements were seen, no conclusive position could be reached due to the limited scope of the evaluation as well as discrepancies between human judgment, and automatic classification results.

One of the main weaknesses of the aforementioned approaches is that they mostly rely on the ability of the methodology to provide new learning instances by propagation within WordNet. However, in all cases, they do not take proper advantage of the relational structure of WordNet. Indeed, semantic coherence (for TWnL and TWnH) is only calculated between new instances and synsets from the seed list, but never between new instances themselves.[3] However, one may expect that highly correlated new instances should be treated commonly. One solution to deal with this problem is to define the classification problem as an optimization process, where both semantic coherence and temporal orientation are treated as combined objectives. For that purpose, we propose to adapt the standard s-t mincut algorithm (Blum and Chawla, 2001) to our particular semi-supervised multi-class learning problem.

3 Learning with s-t mincut

The s-t mincut algorithm is based on finding minimum cuts in a graph, and uses pairwise relationships among examples in order to learn from both labeled and unlabeled data. In particular, it outputs a classification corresponding to partitioning a graph in a way that minimizes the number of similar pairs of examples that are given different labels.

3.1 Main Principles

Let us consider n items $x_1, \ldots x_n$ to divide into two classes C_1 and C_2 based on two different types of information. The first information type – the *individual score* denoted as $ind_j(x_i)$ – measures the non-negative estimate of each x_i belonging to class C_j based on the features of x_i alone. The second information type – the *association score* denoted as $assoc(x_i, x_k)$ – represents the non-negative estimate of how important is that x_i and x_k be in the same class.

This situation can be represented as an undirected graph G with vertices $\{v_1, \ldots, v_n, s, t\}$, where s and t are respectively the *source* and *sink* vertices, which represent each class label and one vertex v_i corresponds to a given item x_i. If s

[3] This may occur only through a side-effect process.

(resp. t) corresponds to class C_1 (resp. C_2), we add n edges (s, v_i), each with weight $ind_1(x_i)$, and n edges (v_i, t), each with weight $ind_2(x_i)$. Finally, we add $\binom{n}{2}$ edges (v_i, v_k), each with weight $assoc(x_i, x_k)$.

The learning process corresponds to finding the minimum cut in G that minimizes some cost function, where (i) a cut (S,T) of G is a partition of its nodes into sets $S = \{s\} \cup S'$ and $T = \{t\} \cup T'$ where $s \notin S'$ and $t \notin T'$, and (ii) its cost $cost(S,T)$ is the sum of the weights of all edges crossing from S to T, as defined in equation (1):

$$\sum_{x \in C_1} ind_2(x) + \sum_{x \in C_2} ind_1(x) + \sum_{x_i \in C_1, x_k \in C_2} assoc(x_i, x_k) \quad (1)$$

3.2 Advantages and Disadvantages

Formulating the task of temporality detection on word senses in terms of graphs allows us to model item-specific and pair-wise information independently. As a consequence, machine learning algorithms representing temporal indicators can be used to derive *individual* scores for a particular sense in isolation. The edges weighted by the *individual* scores of a vertex (sense) to the source/sink can be interpreted as the probability of a sense belonging to a given temporal class without taking into account similarity to other senses.

At the same time, we can use conceptual-semantic relations from WordNet to derive the *association* scores. The edges between two senses weighted by the *association* scores can indicate how similar two senses are. If two senses are connected via a temporality-preserving relation, they are likely to both belong to a temporal class. For instance, hyponymy relation is usually a temporality-preserving relation,[4] where two hyponyms such as *present, nowadays — the period of time that is happening now* and *now — the momentary present* are both temporal.

To detect the temporal orientation of word senses, Dias et al. (2014) and Hasanuzzaman et al. (2014b) adopted a single view instead of two views on the data. The ability to combine two views on the data is precisely one of the strengths of the s-t mincut strategy.

Second, the s-t mincut algorithm is a semi-supervised framework. This is essential as the existing labeled datasets for our problem are small. In addition, glosses are short, leading to sparse high-dimensional vectors in standard feature representations. Furthermore, WordNet connections between different parts of the WordNet hierarchy can be sparse, leading to relatively isolated senses in a graph in a supervised framework. The mincut strategy allows us to import unlabeled data that can serve as bridges to isolated components. More importantly, the unlabeled data can be related to the labeled data (by some WordNet relation) and might help to pull unlabeled data to the right cuts.

It is also important to note that transductive methods such as the s-t mincut algorithm particularly suit our case study as all learning examples are known. However, the addition of new word senses would require the re-application of the method to the entire graph. Indeed, the model does not learn to predict unseen examples.

3.3 Methodology

The formulation of our mincut strategy for temporal classification of synsets involves the following steps.

Step I. We define two vertices s (source) and t (sink), which correspond to the *temporal* and *atemporal* categories, respectively. Vertices s and t are *classification vertices*, and all other vertices (labeled, unlabeled, and test) are *example vertices*.

Step II. The labeled examples are connected to the classification vertices they belong to via edges with high constant non-negative weight. The unlabeled examples are connected to the classification vertices via edges weighted with non-negative scores that indicate the degree of belonging to both the *temporal* and *atemporal* categories. Weights (i.e. individual scores) are calculated based on a supervised classifier learned from labeled examples (cf. Section 3.4).

Step III. For all pairs of example vertices, for which there exists a listed semantic relation in WordNet, an edge is created. This one receives a non-negative score that indicates the degree of semantic relationship between both vertices and corresponds to the association score (cf. Section 3.5).

Step IV. The max-flow theorem (Papadimitriou and Steiglitz, 1998) is applied over the built graph to find the minimum s-t cut.[5]

[4] Although Dias et al. (2014) show that this is not always the case.

[5] Max-flow algorithms show polynomial asymptotic running times and near-linear running times in practice.

Step V. The temporal partition is then divided into three temporal sub-partitions (*past*, *present*, and *future*) following a hierarchical strategy. First, we define two new vertices s and t, which correspond to *past* and *not_past* categories, respectively, and follow steps *II* through *IV*. This divides the subgraph into two disjoint subsets, i.e. *past* synsets, and synsets belonging either to *present* or *future*. Finally, we repeat steps *II* through *IV*, where vertices s and t correspond to *future* and *present*, respectively (cf. Section 3.6).

3.4 Individual Scores

The non-negative edge weights to s and t denote how an example vertex is related to a specific class. For the unlabeled and test examples, a supervised learning strategy is used to assign edge weights. Each synset from a labeled dataset – we use the dataset provided by Dias et al. (2014) – which contains *past*, *present*, *future* and *atemporal* senses is represented by its gloss encoded as a vector of word unigrams weighted by their frequency.[6] Then, depending on the classification task, a two-class SVM classifier is built from the Weka platform.[7] In particular, the SVM membership scores are transformed into probability estimates based on Platt calibration (Niculescu-Mizil and Caruana, 2005), which are directly mapped to edge weights. In Table 1, we present the 10-fold cross-validation results for all classifiers tested in the context of this work.

In order to ensure that the mincut procedure does not reverse the labels of the labeled examples, a high non-negative constant weight of 3 is assigned to any edge between a labeled vertex and its corresponding classification vertex, and a low non-negative constant weight of 0.001 to the edge to the other classification vertex. This is a classical implementation of $+\infty$ and $1/+\infty$ theoretical weights.

3.5 Association Scores

While formulating the graph, we connect two example vertices by an edge if they are linked by one of the 10 WordNet relations presented in Table 2. The main motivation towards using other relations in addition to the most frequently encoded relations (e.g. hypernym/hyponym) among synsets in WordNet is to achieve high graph connectivity.

[6]Other sentence representations could be tested but this is out of the scope of this paper.

[7]http://www.cs.waikato.ac.nz/ml/weka/

Two class problem	Accuracy	F1
temporal vs. *atemporal*	92.3	94.2
past vs. *not_past*	90.4	90.2
present vs. *not_present*	85.3	85.2
future vs. *not_future*	90.1	89.9
present vs. *future*	87.3	86.4

Table 1: SVM results for individual scores.

Wordnet Relation	#same	#different	Weight
Direct-Hyponym	73268	7246	0.91
Similar-to	6587	1914	0.77
Direct-Hypernym	61914	9600	0.76
Attribute	350	109	0.76
Also-see	1037	337	0.75
Troponym	6917	2651	0.72
Derived-from	3630	1947	0.65
Domain	2380	2895	0.45
Domain-member	2380	2895	0.45
Antonym	1905	3614	0.35

Table 2: Association scores with *DiffWt* Method.

Different weights can be assigned to different relations to reflect the degree to which they preserve temporality. Therefore, we adopt two strategies to assign weights to different WordNet relations. The first method (*ScWt*) assigns the same constant weight of 1.0 to all WordNet relations. The second method (*DiffWt*) considers several degrees of preserving temporality. In order to do this, we adopt a simple rule-based strategy to produce a large noisy set of *temporal* and *atemporal* synsets from WordNet. First, we take the list of 30 hand-crafted temporal seed synsets (equally distributed over *past*, *present*, and *future*) proposed in Dias et al. (2014) along with their direct hyponym synsets. This forms a temporal list. Then, each WordNet synset that contains a word sense from the temporal list in its gloss is 'roughly' classified as *temporal*. Otherwise, it is considered as *atemporal*. We then simply count how often two synsets connected by a given relation have the same or different temporal dimension. Finally, the weight is calculated by #same/(#same+#different) and corresponds to the association score between two example vertices. Results are reported in Table 2.

Note that the exact same strategy is used for the two hierarchical steps, for which new association scores are calculated.

3.6 Hierarchical Strategy

The order of the hierarchical process is driven by classifier accuracy over the labeled dataset pro-

vided by Dias et al. (2014) (cf. Section 4). In order to give the maximum chance of good partitioning at the second level of the hierarchy, we choose the classification problem to handle based on the SVM classifier that demonstrates highest accuracy over the following problems: *past* vs. *not_past*, *present* vs. *not_present*, and *future* vs. *not_future*. In so doing, we can rely on the best possible individual score function. As can be seen in Table 1, this is the case for *past* vs. *not_past*, which happens to be the first sub-partitioning problem. The third level is straightforward, i.e. *present* vs. *future*. We are aware that this simple strategy is prone to bias. However, as manual evaluation of the final resource is involved, producing more results was logistically hard to handle. Nonetheless, testing all combinations remains work that needs to be conducted in the future.

4 Datasets

Labeled Dataset. We used a list that consists of 632 *temporal* synsets and an equal number of *atemporal* synsets provided by Dias et al. (2014) as labeled data for our experiments. Temporal synsets are distributed as follows: 210 synsets marked as *past*, 291 as *present*, and 131 as *future*.

Test Dataset. As the labeled dataset is small, we created an annotation task using the CrowdFlower platform[8] in order to produce a testset. For the annotation task, 398 synsets equally distributed over nouns, verbs, adjectives, and adverbs along with their lemmas and glosses were randomly selected from WordNet[9] as representative of the whole WordNet. Note that this number is a statistically significant representative sample of all WordNet synsets calculated as defined in Israel (1992).

The annotators were expected to answer two questions for a given synset (lemmas and gloss were also provided). While the first question is related to the decision as to whether a synset is *temporal* or *atemporal*, the motivation behind the second question is to collect a more fine-grained (*past*, *present*, *future*) gold-standard.[10] The reliability of the annotators was evaluated on 60 control synsets from the labeled dataset, and 10 ambiguous synsets associated to more than one temporal dimension. Similary to Tekiroglu et al. (2014), raters who scored at least 70% accuracy on average on both sets were considered to be reliable. Finally, each synset was annotated by at least 10 reliable raters.

To have a concrete idea about the agreement between annotators, we calculated the majority class for each synset in our dataset. A synset belongs to a majority class k if the most frequent annotation for the synset was selected by at least k annotators. As a consequence, a large percentage of synsets belonging to high majority classes are symptomatic of good inter-annotator agreement. Table 3 shows the observed agreement. Similarly to Özbal et al. (2011), we consider all annotations with a majority class greater than 5 as reliable. In this case, for the *temporal* vs. *atemporal* annotation scheme, 84.83% of the synsets were annotated identically by the majority of annotators, while for *past*, *present*, and *future*, 72.36% of the annotations fell into this case. As such, we can be confident that the annotation process was successful and the dataset is reliable.

5 Intrinsic and Extrinsic Evaluations

Different intrinsic and extrinsic evaluations have been proposed in prior studies. We compare our work to the same tasks as proposed by Dias et al. (2014) and Hasanuzzaman et al. (2014b), and introduce an extra experiment on temporal relation annotation.

5.1 Inter-Annotator Agreement

In order to compare our approach to prior works, we adopted a similar evaluation strategy as proposed in Dias et al. (2014) and Hasanuzzaman et al. (2014b). To assess human judgment regarding the temporal parts, inter-rater agreement with multiple raters (i.e. 3 human annotators with the 4th annotator being the classifier) was performed over a set of 398 randomly selected synsets. The free-marginal multirater kappa (Randolph, 2005) and the fixed-marginal multirater kappa (Siegel and Castellan, 1988) values are reported in Table 4 and assess moderate agreement for previous versions of TempoWordNet (TWnL, TWnP and TWnH), while good agreement is obtained for the resources constructed by mincuts with both *ScWt* (MC1) and *DiffWt* (MC2) weighting schemes. Note that slightly different results than the ones reported by

[8] http://www.crowdflower.com/

[9] WordNet version 3.0 was used and all sysnsets were selected outside the labeled dataset.

[10] Details of the annotation guidelines are out of the scope of this paper.

Majority Class	3	4	5	6	7	8	9	10
Synset as *temporal* or *atemporal*	0.20	1.21	4.32	10.69	14.56	29.34	19.23	11.01
Temporal synset into *past*, *present*, or *future*	1.23	3.01	10.45	20.22	16.56	12.34	14.23	9.01

Table 3: Percentage of synsets in each majority class.

Hasanuzzaman et al. (2014b) are seen as the number of annotated synsets is much bigger in our experiment (398 instead of 50). These agreement values provide a first and promising estimate of the improvement over the previous versions of TempoWordNet. We plan to confirm that in the future by comparing the systems to a true reference instead of observing the agreement between the systems and a multi-reference as we currently do.

Metric	TWnL	TWnP	TWnH	MC1	MC2
Fixed-marginal κ	0.51	0.46	0.54	0.71	0.78
Free-marginal κ	0.52	0.55	0.59	0.85	0.86

Table 4: Inter-annotator agreement.

5.2 Word Sense Classification

In order to compare our semi-supervised mincut approach to a reasonable baseline, we use a rule-based approach to classify test data into *past*, *present*, *future*, or *atemporal* categories. First, time expressions in glosses are identified and resolved via SUTime tagger (Chang and Manning, 2012). Then, for each synset, its time tags (e.g. FUTURE_REF) are considered as the temporal class for that particular synset. In cases where more than one temporal expression was observed (which occurred in less than 1% of the cases), the majority class is selected. If no time expression is identified by the time tagger, the list composed of 30 hand-crafted temporal seeds proposed in Dias et al. (2014) along with their direct hyponyms and a given list of standard temporal adverbials, prepositions and adjectives are used to classify synsets with one *temporal* dimension or *atemporal*. The performance of this simple rule-based approach is measured for the test data and presented in Table 5 as the baseline configuration. Note that to figure out the contribution of word sense disambiguation, the classical Lesk algorithm (Lesk, 1986) was used to choose the right sense for a given word instead of the most frequent sense. We found that this contribution is negligible (< 0.4% improvement in accuracy).

Comparative results are also presented against prior works: TWnL, TWnP, and TWnH. Table 5 shows that our configurations (MC1, MC2) perform significantly better than previous approaches. In particular, they achieve highest accuracies for *temporal* vs. *atemporal* and *past*, *present*, *future* classifications with improvements of 11.3% and 10.3%, respectively, over the second-best strategy, namely TWnH. Note that this enhancement is mainly due to higher precision overall.

Different training data sizes. In order to better understand the importance of the size of labeled data in the context of semi-supervised classification strategies, we propose the following experiments.

We randomly generate equally distributed subsets of training data L_i (from a set of 632 *temporal* and 632 *atemporal* synsets) such that $L_1 \subset L_2 \subset L_3 \ldots \subset L_n$. For each labeled dataset, we run the mincut strategy with *DiffWt* (i.e. MC2) and compare it to the hybrid propagation proposed by Hasanuzzaman et al. (2014b) (i.e. TWnH). Accuracies of both approaches over the test data are presented in Table 6.

The s-t mincut approach performs consistently better than the propagation strategy. In particular, we show that with 400 labeled examples better results can be obtained than relying on 1264 training items within a propagation paradigm.

Considering the above findings, we selected the MC2 configuration obtained with maximum labeled data for the extrinsic experiments, which includes 110,002 atemporal synsets, 1733 past synsets, 4193 present synsets, and 1730 future synsets.

5.3 Temporal Sentence Classification

Temporal sentence classification has traditionally been used as the baseline extrinsic evaluation and consists of labeling a given sentence as *past*, *present* or *future*. In order to produce comparative results with prior works, we test our methodology on the balanced dataset produced in Dias et al. (2014), which consists of 1038 sentences equally distributed as *past*, *present* and *future*.

Moreover, we propose to extend these experiments with a corpus of 300 temporal posts from

Method	Baseline	TWnL	TWnP	TWnH	MC1	MC2
Accuracy	48.8	65.6	62.0	68.4	74.4	**79.7**
temporal (p, r, f1)	(52.0, 56.3, 54.0)	(63.5, 82.1, 71.6)	(55.8, 84.2, 67.1)	(67.4, 81.9, 73.9)	(84.5, 79.8, 82.0)	(89.1, 79.3, 83.9)
atemporal (p, r, f1)	(58.2, 54.2, 56.1)	(68.3, 79.2, 73.3)	(58.9, 75.6, 66.2)	(69.3, 82.6, 75.3)	(81.3, 86.6, 83.8)	(87.4, 90.8, 89.1)
Accuracy	45.6	62.0	59.6	65.7	72.7	**76.0**
past (p, r, f1)	(49.3, 46.7, 47.9)	(61.2, 73.0, 66.5)	(59.3, 79.1, 67.7)	(63.1, 75.0, 68.0)	(71.1, 79.5, 75.0)	(81.2, 78.5, 79.8)
present (p, r, f1)	(55.3, 48.2, 51.5)	(63.0, 75.2, 68.5)	(58.0, 78.2 66.0)	(77.4, 69.2, 73.0)	(73.0, 71.5, 72.2)	(85.1, 74.7, 79.0)
future (p, r, f1)	(48.5, 49.0, 48.7)	(62.1, 71.9, 66.6)	(57.0, 83.1, 67.6)	(60.0, 75.6, 66.8)	(79.4, 69.5, 74.0)	(86.1, 70.0, 77.2)

Table 5: Accuracy for *temporal vs. atemporal* and *past, present, future* classifications using different methods measured over test data. Results are broken down by precision (p), recall (r), and f1-measure (f1) scores.

Twitter. This corpus contains 100 tweets for each temporal class, which have been time-tagged using the CrowdFlower platformḟootnoteAnnotation details are out of the scope of this paper. For both experiments, each sentence/tweet is represented as a semantic vector space model in the exact same way as proposed in Dias et al. (2014). Thus, a given learning example is a feature vector, where each attribute is either a unigram or a synonym of any temporal word contained in the sentence/tweet and its value is the tf.idf. Note that word sense disambiguation is performed using the Lesk algorithm (Lesk, 1986).

Amount of labeled data	TWnH	MC2
100	59.8	64.3
200	62.6	67.5
400	65.5	73.7
600	67.4	77.6
800	67.9	79.2
1000	68.0	79.0
1264 (all)	68.4	79.7

Table 6: Accuracy results with different sizes of labeled data for *temporal* vs. *atemporal* classification.

Comparative classification results are reported in Table 7 and show small improvements in the mincut strategy, when compared to propagation strategies. In particular, for tweet classification, TWnP shows similar results mainly due to its large coverage of temporal senses (counterbalanced by low precision as confirmed by Table 5). Indeed, TWnP contains 53,001 temporal synsets while MC2 only has 7656 temporal synsets. Note that the semantic enhancement is limited only to the synonymy relation, which drastically restricts the benefit of the semantic vector space model and due to the limited number of analyzed sentences/tweets, huge improvements were not expected.

5.4 Temporal Relation Annotation

Finally, we focus on the problem of classifying temporal relations as proposed in TempEval-3, assuming that the identification of events and timexes is already performed.

In order to produce comparative results with the best-performing system at TempEval-3, namely UTTime (Laokulrat et al., 2013) for the above task, we follow the guidelines and use the same datasets provided by the organizers (UzZaman et al., 2013).

In particular, we restrict our experiment to a subset of relations, namely BEFORE (*past*), AFTER (*future*), and INCLUDES (*present*), with all other relations mapped to the NA−RELATION for the following two subtasks: *event to document creation time* and *event to same sentence event*. This choice is motivated by the complexity of mapping the 14 relations of TempEval-3 into three temporal classes (*past*, *present*, *future*). As such, we test a simpler configuration of the original problem, but we do expect to draw conclusive remarks as minimum bias is introduced.

Note that the underlying idea of this evaluation is to measure the intuition expressed by (Kuzey et al., 2016) that temporal information extraction systems may benefit from the existence of temporal resources. If this is confirmed, deeper research should be conducted to adequately use such a proposed temporal resource for the whole task.

To solve this classification problem, we adopt a simple supervised learning strategy based on state-of-the-art characteristics, plus features from a time-augmented version of WordNet. In particular, each pair of entities to be classified as BEFORE, AFTER, INCLUDES or NA-RELATION is encoded with the following features:

- String features: the tokens and lemmas of each entity pair;

- Grammatical features: the part-of-speech tags

Method	TWnL	TWnP	TWnH	MC2
Sentence classification (p,r,f1)	(69.7,66.1,66.7)	(68.2,70.5,69.3)	(69.8,67.6,68.6)	(73.3,70.1 **71.4**)
Tweet classification (p,r,f1)	(51.4,47.1,49.1)	(50.4,52.8,51.5)	(51.8,48.2,49.8)	(52.8,50.6,**51.6**)

Table 7: Results for temporal sentence and tweet classification performed on 10-fold cross validation with SVM with Weka default parameters.

of the entity pair (only for event-event pairs), and a binary feature indicating whether the entity pair has the same PoS tag;

- Entity attributes: the entity pair attributes as provided in the dataset. These include class, tense, aspect, and polarity for events, while the attributes of time expressions are its type, value, and dct (indicating whether a time expression is the document creation time or not);

- Dependency relation: the type of dependency and the dependency order between entities;

- Textual context: the textual order of the entity pair;

- Temporal lexicon: the relative frequency of each temporal category (*past*, *present*, *future*) appearing in the context of an entity pair; the context is considered as (i) the text appearing between entities, (ii) the text of all tokens in a time expression, and (iii) 5 tokens around time expressions or events. The features are encoded as the frequency with which a word from a temporal category appeared in the text divided by the total number of tokens in the text.

Approaches	Precision	Recall	F1
UTTime	57.5	58.7	58.1
TRMC2	**66.9**	**68.7**	**67.7**
TRTWnH	61.2	62.5	61.8

Table 8: Temporal relation classification results.

Based on this feature representation, the two best classifiers for *event to document creation time* and *event to same sentence event* subtasks are selected via a grid search over parameter settings. The grid is evaluated with a 5-fold cross validation on the training data and SVM classifiers are chosen with default parameters of the Weka platform. This produces two systems, namely TRMC2 and TRTWnH depending on the temporal lexicon used: MC2 or TWnH. Note that we also measure the performance of UTTime for the settings stated above.

Table 8 presents comparative evaluations. Results show that TRMC2 outperforms all other approaches and achieves highest performance in terms of precision, recall, and F1-measure. However, more important still is the fact that a simple learning strategy with some temporal lexicon (MC2 or TWnH) leads to improved results, when compared to some solution that does not take advantage of such a resource (UTTime, here).

Features	F1	Features	F1
mfc baseline	33.55	all features	**67.7**
string alone	45.06	w/o *string*	65.70
grammatical alone	46.96	w/o *grammatical*	64.85
entity alone	52.23	w/o *entity*	62.08
dependency alone	48.65	w/o *dependency*	65.06
textual alone	46.82	w/o *textual*	64.96
temporal alone	51.62	w/o *temporal*	62.76

Table 9: Feature ablation analysis. The most frequent class baseline (mfc).

In order to measure the real impact of the temporal lexicon features, we present feature ablation analyses in Table 9. Results clearly show the importance of the features based on the temporal lexicon, being the second best-performing feature set. As a consequence, we may conclude that improvements in temporal analysis may be obtained by the correct use of some temporal lexical resource.

6 Conclusions

In this paper, we proposed a semi-supervised min-cut strategy to address the relatively unexplored problem of associating word senses with their underlying temporal dimensions. We produce a reliable temporal lexical resource by automatically time-tagging WordNet synsets into *past*, *present*, *future* or *atemporal* categories. The underlying idea is that instead of using a single view on the data (as done in prior work), multiple views result in better temporal classification accuracy. In particular, both intrinsic and extrinsic experimental results confirm the soundness of the proposed approach and support our initial hypotheses. Note that the all resources created within this work are publicly available.

Acknowledgments

The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Avrim Blum and Shuchi Chawla. 2001. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 19–26, Massachusetts, USA.

Ricardo Campos, Gaël Dias, Alípio M. Jorge, and Adam Jatowt. 2014. Survey of temporal information retrieval and related applications. *ACM Computing Survey*, 47(2):15:1–15:41.

Angel X. Chang and Christopher Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, pages 3735–3740, Istanbul, Turkey.

Gaël Dias, Mohammed Hasanuzzaman, Stéphane Ferrari, and Yann Mathet. 2014. Tempowordnet for sentence time tagging. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion (WWW)*, pages 833–838, Seoul, South Korea.

Mohammed Hasanuzzaman, Gaël Dias, Stéphane Ferrari, and Yann Mathet. 2014. Propagation strategies for building temporal ontologies. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 6–11, Gothenburg, Sweden.

Erdal Kuzey, Jannik Strötgen, Vinay Setty, and Gerhard Weikum. 2016. Temponym tagging: Temporal scopes for textual phrases. In *Proceedings of the 25th International Conference Companion on World Wide Web (WWW)*, pages 841–842, Montreal, Canada.

Natsuda Laokulrat, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. Uttime: Temporal relation classification using deep syntactic features. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 88–92, Atlanta GA, USA.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Inderjeet Mani, James Pustejovsky, and Robert Gaizauskas. 2005. *The language of time: a reader*, volume 126. Oxford University Press.

Georges A. Miller. 1995. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 625–632, New York, NY, USA. ACM.

Christos H. Papadimitriou and Kenneth Steiglitz. 1998. *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

Justus J. Randolph. 2005. Free-marginal multirater kappa (multirater κfree): an alternative to fleiss' fixed-marginal multirater kappa. *Joensuu Learning and Instruction Symposium*.

H. Andrew Schwartz, Greg Park, Maarten Sap, Evan Weingarten, Johannes Eichstaedt, Margaret Kern, Jonah Berger, Martin Seligman, and Lyle Ungar. 2015. Extracting human temporal orientation in facebook language. In *Proceedings of The 2015 Conference of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies (NAACL)*, Denver, Colorado, USA.

Sydney Siegel and John Castellan. 1988. *Nonparametric Statistics for the Social Sciences*. Mcgraw-hill edition.

Jannik Strötgen and Michael Gertz. 2015. A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 541–547, Lisbon, Portugal.

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 *Proceedings of 2nd Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 1–9, Atlanta, Georgia, USA.