

Extending Neural Temporal Tagging Systems with External Knowledge



Martin Pömsl

Dr. phil. Tobias Thelen

Institute of Cognitive Science, Osnabrück

M. Sc. Luzian Hahn

Fraunhofer Institute for Integrated Circuits, Erlangen

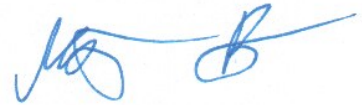
Osnabrück University

This thesis is submitted for the degree of
Bachelor of Science

February 5th, 2021

Declaration of Authorship

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.



Martin Pömsl
Osnabrück, February 5th, 2021

Abstract

Temporal tagging is the task of recognizing temporal expressions and normalizing them to a common machine-readable format. The task is of great importance in situations where machines must understand the time that a human is referring to, such as scheduling an event with a virtual assistant or analyzing news. In the past, systems that rely on hand-crafted rules have shown the best performance at solving this task for news articles. However, since rule-based systems only perform well for the domains and languages that they have been designed for, it would be desirable to have a domain-adaptive data-driven system that performs equally well.

One reason why rule-based systems currently perform better than data-driven systems might be that the humans that formulate the rules of rule-based systems inject some of their knowledge about the world and the human calendar into the system. This work investigates whether giving data-driven neural systems access to the same kind of general and temporal knowledge can help them perform as well as rule-based systems.

The experimental results indicate that for the news domain, the performance of neural systems on the subtask of temporal expression recognition can be improved slightly by augmenting them with external knowledge. However, the improved performance is still below that of the best rule-based systems. For the domain of virtual assistant commands, the neural systems are on their own already much better at recognizing temporal expressions than existing rule-based systems since they can adapt to the specific properties of the domain. The experiments showed no further improvements through knowledge augmentation for the voice assistant commands domain.

In order to evaluate the full task of both recognizing and normalizing temporal expressions, the neural systems are combined with an existing rule-based normalization system. In this setting, neural systems are firmly outperformed by existing rule-based systems, as were all other systems that rely on this particular rule-based normalization system. The results indicate that in order to translate improvements through knowledge augmentation on the subtask of temporal expression recognition to the full task of temporal tagging, there is the need for a data-driven normalization system that can learn to normalize those expressions that can only be identified with the help of external knowledge.

Table of contents

1	Introduction	1
2	Related Work	4
2.1	Theoretical Background	4
2.2	Temporal Tagging	8
2.3	Knowledge Augmentation	15
3	Resources	17
3.1	Data Sets	17
3.2	Knowledge Sources	19
4	Methodology	22
4.1	Neural Temporal Expression Recognition	22
4.2	Knowledge-augmented Neural Temporal Expression Recognition	28
4.3	Evaluation	31
5	Experimental Setup	33
5.1	Systems	33
5.2	Experiments	35
6	Results	37
6.1	Neural Temporal Expression Recognition	37
6.2	Knowledge-augmented Neural Temporal Expression Recognition	38
6.3	End-to-end Temporal Tagging	40
7	Discussion	42
7.1	Neural Temporal Expression Recognition	42
7.2	Knowledge-augmented Neural Temporal Expression Recognition	44
7.3	End-to-end Temporal Tagging	46
8	Conclusion	49

Chapter 1

Introduction

For as long as computers have existed, humans have been trying to find ways to communicate their desires to computers in a way that is as effortless as possible for humans and as unambiguous as possible for computers. This quest has led to the research field of natural language understanding, which aims to devise systems that can operate on text in one of the human languages and solve tasks related to the meaning conveyed in the text.

Temporal tagging is the task of recognizing temporal expressions such as "on Tuesday, 26th of January 2021" in a text and normalizing them to a common machine-readable format such as "2021-01-26". In the shared task series TempEval (Verhagen et al., 2010; UzZaman, Llorens, et al., 2013) and subsequent publications that evaluate on the TempEval-3 data set, rule-based systems were established to perform best at this task, with the best recognition system SynTime (Zhong, Sun, and Cambria, 2017) being fully rule-based. The best system for the full task of both recognizing and normalizing temporal expressions, also called end-to-end temporal tagging, is the hybrid system UWTime (Lee et al., 2014), which relies on a combination of hand-crafted rules and learned parameters. Fully data-driven systems that either use traditional machine learning methods such as logistic regression and support-vector machines (Bethard, 2013) and recent neural methods (Etcheverry and Wonsever, 2017; Lange et al., 2020) consistently report worse results than their rule-based counterparts for Temporal Expression Recognition (TER) on the TempEval-3 data set. For temporal expression normalization, no fully data-driven system has been proposed yet, which reflects the difficulty of predicting the correct fine-grained string from the large output space of all possible points in time with current machine learning methods.

One possible explanation for the inferiority of data-driven systems is that the hand-annotated data sets for temporal tagging are small in size compared to other natural language understanding tasks on which data-driven systems have been successful so far (Wang et al., 2018). Consequently, the denoted time of named temporal expressions such as obscure

holidays or named events is difficult to learn for supervised data-driven systems, since these expressions will occur only few times or not at all in the train data (Brucato et al., 2013). Rule-based systems solve this by either directly incorporating human knowledge about these expressions into their hand-crafted rules or building such rules automatically from dedicated human-generated resources (Kuzey, Strötgen, et al., 2016).

In order to level the playing field for neural systems, this work aims to extend neural temporal tagging systems with the same kind of knowledge that existing rule-based systems have access to. Knowledge augmentation for neural systems has already proven successful in a number of natural language understanding tasks, including Named Entity Recognition (NER) (Seyler et al., 2018), which is closely related to TER in that they are both sequence labeling tasks.

Apart from the ability to handle rare phenomena in low-resource settings, knowledge augmentation for neural networks promises many other advantages. One of the greatest benefits of extending subsymbolic neural networks with a symbolic external knowledge base is that it increases both interpretability and adaptability. Explicit knowledge augmentation makes it possible for humans to directly observe which information the system has access to and, crucially, to modify this information without having to retrain the whole model. This opens up new applications for neural systems, such as tailoring systems to fit a specific domain or even a specific person by simply modifying the knowledge base accordingly.

Perhaps the most plausible application of this technology is improving the virtual personal assistants that have become more and more common in day-to-day life. If given access to the personalized knowledge base of a user e.g. in the form of an online calendar, a knowledge-augmented TER system could also normalize temporal expressions that have a specific meaning only to a single user, such as planned events or birthdays of family members.

As a first step towards that goal, this work presents a baseline Neural Temporal Expression Recognition (NTER) system and proposes a Knowledge-Augmented NTER (KANter) system that can take into account different kinds of external knowledge when making predictions. For now, the subtask of normalization is left to an existing rule-based system, which makes the resulting end-to-end system a hybrid of data-driven recognition and rule-based normalization.

The remainder of this work is structured as follows: Chapter 2 gives an overview of existing data sets and systems for temporal tagging and briefly summarizes key concepts used to tackle the task of temporal tagging and knowledge augmentation for neural systems in general. In Chapter 3, those data sets that were used in the experiments are introduced in more detail along with the external knowledge sources that were used in the experiments. Chapter 4 first presents the baseline NTER system and then proposes modifications that allow

for the integration of external knowledge. It also details the evaluation protocol used to score the performance of TER and end-to-end temporal tagging systems. Chapter 5 outlines the structure and motivation of the experiments, which aim to evaluate the proposed (KA)NTER methods, replicate the reported results of existing systems and compare all systems on the task of end-to-end temporal tagging in two different domains. In Chapter 6, the results of these experiments are presented, which are then interpreted in Chapter 7 along with qualitative analyses of notable phenomena. Finally, Chapter 6 puts the findings into perspective and highlights further research opportunities resulting from this work.

Chapter 2

Related Work

The following will give a brief overview of the general strategies and recent advances in word representations and sequence labeling necessary to understand the methods employed in this work. After that, the task of temporal tagging will be introduced along with existing systems. Finally, approaches to knowledge augmentation for neural systems in general and for temporal tagging in particular will be discussed.

2.1 Theoretical Background

Temporal tagging systems operate on sequences of words and as such heavily rely on methods from the research areas of sequence labeling and word representation learning.

Sequence Labeling

Sequence labeling is one of the most common task formats in natural language understanding. At the most basic level, sentence splitting, phrase chunking and POS tagging can be cast as the task of labeling each element in a sequence of words. More complex instances of sequence labeling tasks that require reasoning over the semantics of a subsequence of words are e.g. NER and TER.

Label Sets

Sequence labeling systems assign one out of a set of predefined labels to each element in a sequence. For tasks that require labels for subsequences of elements, such as multi-word expressions, the labels can be formulated in such a way that they imply membership in a subsequence of elements of a certain type. Popular choices for such predefined categories are

the BIO label set and the BILOU label set. BIO provides three categories for each expression type, marking the **B**eginning, **I**nside and **O**utside of an expression. BILOU is more fine-grained and provides five categories for each expression type, marking the **B**eginning, **I**nside, **L**ast and **O**utside of an expression as well as **U**nit-length expressions. An instance of a BIO-labeled word sequence for NER can be found in Example 2.1.

Both the BIO and the BILOU label sets have been applied with great success, depending on the requirements of the specific task (Ratinov and Roth, 2009; Reimers and Gurevych, 2017b).

Conditional Random Fields

Conditional Random Fields (CRFs) (Lafferty, McCallum, and Pereira, 2001) are a family of statistical methods that models probabilities over elements in a general graph. Since sequences are a special type of graphs, a variant of CRFS called linear-chain CRFs can be used for a sequence labeling task. Linear-chain CRFs can be seen as an extension of Hidden Markov Models (HMMs), which means they rely on the assumption that each label depends only on the current element and the previous label (Bishop, 2006). Especially in the case of BIO-style labels, this is better than just predicting each label given only its element, since it enables the model to determine if the current element is part of a subsequence that was previously started with a "B-" label or continued with an "I-" label (Lample et al., 2016). However, linear-chain CRFs are better suited for many natural language understanding tasks than HMMs, since they can operate on non-independent features, while HMMs are limited to independent features (Klinger and Tomanek, 2007).

In any sequence labeling problem with a fixed set of labels, the number of possible label sequences rises exponentially with the length of the sequence. When trying to find the most probable label sequence for a given sequence of elements, this can lead to a prohibitively long runtime for naive enumeration methods. For that reason, the most probable labeling sequence in CRFs is usually determined using the Viterbi algorithm, which sequentially determines the optimal next label only in the local bigram context of all possible predecessor labels and thus scales linearly with the sequence length. The process of finding the most probable

Example 2.1 Text with BIO labels for NER. "PER" stands for subsequences of type person, "LOC" for subsequences of type location.

Mark	Watney	visited	Mars	in	the	novel	<i>The</i>	<i>Martian</i>	.
B-PER	I-PER	O	B-LOC	O	O	O	O	O	O

sequence is called decoding and can be achieved by tracing the optimal path backwards using the computed optimal bigram transitions (Bishop, 2006).

Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a family of artificial neural networks that, similar to CRFs, model the transitions between elements of a sequence. While computationally similar to Feedforward Neural Networks (FFNNs), RNNs consume sequences one element at a time and keep a hidden state h which is updated at each element by multiplication with a learned matrix of real-valued weights (Jurafsky and Martin, 2009). Either the sequences of hidden states or the last hidden state are taken to be the prediction of the model, depending on whether a sequence prediction or an aggregation prediction is desired. Similar to HMMs, the hidden state of a RNN at one time step depends only on the previous hidden state and the current input.

RNNs are generally trained using a modified version of error backpropagation called backpropagation through time. However, the propagation of gradients through longer sequences of elements inherently suffers from the problem of vanishing and exploding gradients. Gradients are said to be vanishing if they get very small because of repeated multiplication during backpropagation, and they are said to be exploding if they get very large because of repeated multiplication during backpropagation (Hochreiter, 1998).

Long Short-Term Memory units (LSTMs) (Hochreiter and Schmidhuber, 1997) have to some extent solved this problem by truncating gradients and regulating the flow of information through specialized gates, which decide when to store and forget information. For this reason, LSTMs are currently one of the most widely used methods for sequence labeling in natural language understanding (Lample et al., 2016). Another type of RNN that is becoming more and more popular in natural language understanding are Gated Recurrent Units (GRUs) (Cho et al., 2014), which rely on fewer gates than LSTMs, making them less complex and more computationally efficient.

Word Representations

In order to apply numerical data-driven methods to text data, it is necessary to represent words with meaningful features. These features may either be categorical or continuous in nature.

Syntactic Word Representations

Many machine learning systems and formalisms use syntactic features to categorize the atomic words that make up a given text, which are also called tokens (Jurafsky and Martin, 2009). Syntactic features that are often used to characterize tokens include:

- Part-of-Speech (POS): Lexical category of word in context
- Dependency: Syntactical relations between words
- Stem: Inflectional morphological root

For the use in neural networks, which operate on real-valued inputs, categorical features such as POS may be either be encoded to one-hot vectors or embedded in a learned randomly-initialized lookup matrix of real-valued vectors.

Semantic Word Representations

Semantic words representations aim to reflect the meaning of words. The prevailing standard for semantic word representations is based on the distributional hypothesis, the notion that the meaning of a word can be inferred from its neighboring words (Harris, 1954). Following this idea, distributional semantic representations in the form of word vectors can be derived from word co-occurrence statistics.

Next to counting how often words occur next to each other, another way to induce word vectors is predicting which words occur next to each other. In this method, a basic single-layer Feed-Forward Neural Network (FFNN) is trained on the task of predicting neighboring words. The real-valued intermediate word representations that are learned in that process are called neural word embeddings and have proven useful for many natural language understanding tasks. Word2vec (Mikolov et al., 2013) was the first efficient method to induce such word representations.

Building on that approach, a number of similar neural word embedding strategies have since been devised. GloVe vectors (Pennington, Socher, and C. Manning, 2014) leverage both statistics and neural methods by taking into account not only the local neighborhood of a word, but also the global neighborhood. FastText vectors (Bojanowski et al., 2017) work in principle like the neural embeddings presented by Mikolov et al., but also take into account the internal structure of words, which is important for languages that rely heavily on morphology.

In recent years so-called contextualized word embeddings, which represent the meaning of a word use within a certain context, rather than independent from its context, have delivered promising results. These contextualized word representations are often produced by training

a neural language model, whose goal it is to predict unknown tokens from a sequence of known tokens (Jurafsky and Martin, 2009).

ELMo embeddings (M. Peters et al., 2018) consist of learned linear combinations of word representations derived from a language model based on multiple stacked bidirectional LSTMs. FLAIR embeddings (Akbik, Blythe, and Vollgraf, 2018) achieve contextualization by training a character-level LSTM language model and concatenating the hidden states of the characters at the beginning and end of a word.

Apart from LSTMs, the Transformer architecture (Vaswani et al., 2017) has gained notoriety as an effective architecture for neural language models. This architecture is mainly based on stacking multiple instances of self-attention on top of each other. The core idea of the basic attention mechanism is that elements in one sequence are aggregated by multiplying them with normalized scalars derived from another sequence of elements (Bahdanau, Cho, and Bengio, 2015). If the scalar associated with an element is high, then the system is said to pay attention to an element, since the result of the operation retains a large part of its content. In the case of self-attention, both sequences are identical.

As opposed to RNNs, self-attention directly relates each element in a sequence to each other element in the sequence. This has the advantage that long-range dependencies between elements are easier to resolve than in RNNs, where information can only flow through the hidden states of the elements in between. However, operating on all pairs of elements also comes with a runtime complexity that scales quadratically rather than linearly with the sequence length. This issue can be partly alleviated by the fact that the computation of attention for any one element to all other elements is predefined and not dependent on the results of the attention computations of other elements, which makes the attention mechanism easy to parallelize via matrix multiplication (Vaswani et al., 2017).

One of the most widely used models that successfully utilize the Transformer architecture with a language modeling objective to provide useful contextualized word embeddings is BERT (Devlin et al., 2019), whose pretrained representations can be used to achieve state-of-the-art performance for numerous natural language understanding tasks.

2.2 Temporal Tagging

Temporal tagging is the task of recognizing and normalizing temporal expressions that occur in a text. Temporal expressions are words or subsequences of words that refer to a certain time. This definition of temporal tagging was established in the temporal shared task series TempEval, the most recent installments of which are TempEval-2 (Verhagen et al., 2010) and TempEval-3 (UzZaman, Llorens, et al., 2013).

Temporal tagging is often segmented into the natural subtasks of Temporal Expression Recognition (TER) and normalization. After temporal expressions have been recognized, e.g. with a sequence labeling model predicting BIO labels, they have to be normalized to a common time format. The desired output depends on the annotation schema that is used. The TempEval series uses the XML-based TimeML TIMEX3 standard, which is based on ISO 8601 (J. Pustejovsky et al., 2003). TIMEX3 requires the following annotations for a temporal expression:

- EXTENT: Index of beginning and ending character of a temporal expression
- TYPE: Type of temporal expression (one of TIME, DATE, SET, DURATION)
- VALUE: String denoting time depending on type (e.g. "2004-11-22" for DATE)

The exact allowed VALUE strings for each TYPE of expression can be found in the TimeML annotation guide (Saurí et al., 2006).

A TER-only system provides the extent of an expression, but not the value. A normalization-only system outputs a value string, but relies on already existing extent annotations. Both kinds of systems can additionally predict the type of an expression, however this is not always the case. As a consequence, not all recognition systems and normalizations systems can be combined to yield a working end-to-end temporal tagging system.

Existing Data Sets

There exist a wide variety of annotated data sets for the purpose of temporal tagging, an overview of which can be found in Table 2.1. Annotated data sets exist for multiple languages, however, as is often the case for natural language resources, the most common language is English, followed by German and Spanish. There are three main annotation schemas for temporal tagging that are currently in use: TIMEX3 (J. Pustejovsky et al., 2003), its defunct predecessor TIMEX2 and SCATE (Bethard and Parker, 2016). TIMEX3 and TIMEX2 are older and more established, consequently most data sets use it for annotation and most systems use it as the target format of normalization. It has been argued that SCATE is more comprehensive and better suited to machine learning, since it models temporal expressions as compositions of time entities, which allows for more coarse-grained output values than the TIMEX3 VALUE attribute (Laparra, Xu, Elsayed, et al., 2018). However, SCATE has not received widespread adoption in the community in the years since its conception in 2016, as is evident by the fact that in 2020, only one available data set is using it.

News

The domain in which temporal tagging systems are predominantly evaluated is news. News is a particularly fruitful domain for temporal tagging, since it contains many precisely specified temporal expressions referring to current events. Another plus is that the Document Creation Time (DCT) is usually available, which makes inferring the normalized values for expressions like "today" feasible.

TimeBank and AQUAINT (James Pustejovsky et al., 2003) are two English-language data sets that contain documents mainly from newswire sources such as *Associated Press*, transcribed broadcasts from organizations such as *CNN* and articles from *Wall Street Journal*. The texts were annotated with TIMEX3 tags as well as annotations for other temporal tasks such as temporal relation classification.

The TempEval-2 and TempEval-3 shared tasks are a major source of TIMEX3 annotated news data sets in different languages. The TempEval-2 data set incorporates a subset of 62k annotated words from TimeBank, as well as annotated texts in Chinese, Italian, French, Korean, and Spanish (Verhagen et al., 2010). Building on that, the English portion of TempEval-3 reused a subset of the TempEval-2 data set and in addition added 33k words from AQUAINT. They also annotated a number of texts from the same domain with two different methods: 6k words were annotated manually by expert annotators, which the authors call TempEval-3 Platinum. 666k words from the newswire corpus Gigaword (Napoles, Gormley, and Durme, 2012) were annotated automatically with a weighted ensemble of temporal tagging systems that were successful in previous competitions. This data set is

Table 2.1 Properties of existing temporal tagging data sets. It can be seen that most existing data sets are in English and annotated with the TIMEX3 schema.

	Domain	Language	Annotation Schema	Tokens
TempEval-2	News	English	TIMEX3	63k
TempEval-3	News	English	TIMEX3	768k
TempEval-3	News	Spanish	TIMEX3	67k
KRAUTS	News	German	TIMEX3	75k
PNT	News	English	SCATE	< 95k
WikiWars	Historical Narrations	English	TIMEX2	120k
WikiWarsDE	Historical Narrations	German	TIMEX2	95k
PÂTÉ	Voice Assistant Commands	English	TIMEX3	5k
Tweets	Social Media	English	TIMEX3	20k

called TempEval-3 Silver. In addition to these English data sets, TempEval-3 also provided the opportunity to train and test systems on annotated Spanish data sets (UzZaman, Llorens, et al., 2013).

KRAUTS (Strötgen, Minard, et al., 2018) is a German-language data set that contains 75k tokens from the German newspapers *Zeit* and *Dolomiten*. It was manually annotated with TIMEX3 tags.

Parsing Time Normalizations (PNT) (Laparra, Xu, Elsayed, et al., 2018) is a subset of 968 temporal expressions from the English TempEval-3 data set, that is annotated with temporal entities as specified in the SCATE schema (Bethard and Parker, 2016). In addition to that, it also contains data from the clinical domain.

Historical Narrations

WikiWars (Mazur and Dale, 2010) is an English-language data set that contains TIMEX2 annotations for 22 Wikipedia articles about the course of various historical wars. In total, it contains about 120k annotated tokens. Its German-language counterpart, WikiWarsDE (Strötgen and Gertz, 2011), contains about 95k tokens with TIMEX2 annotations.

Voice Assistant Commands

PÂTÉ (Zarcone, Alam, and Kolagar, 2020) is an English-language data set that contains about 5k tokens with 767 TIMEX3-annotated temporal expressions. The texts were crowdsourced from voice assistant commands given in response to a scenario of an in-car intent such as the rescheduling of an event. The resulting texts were annotated with TIMEX3 tags by an expert.

Social Media

Tweets (Zhong, Sun, and Cambria, 2017) is a manually annotated English-language data set derived from 942 Twitter posts. Due to its social media provenance, the data set contains many irregular grammatical structures and the texts typically exhibit a preference for minimum-length words and abbreviations. The tweets were manually annotated with TIMEX3 tags.

Existing Systems

The SemEval shared task series TempEval (Verhagen et al., 2010; UzZaman, Llorens, et al., 2013) served as a catalyst for the efforts to develop reliable temporal tagging systems and provided a clear-cut task definition as well as a well-established benchmark data set for temporal tagging. Both in the shared tasks and in their aftermath, numerous models were

proposed to address the task of temporal tagging from many different perspectives. The following represents an introduction to the most popular rule-based and data-driven systems that were evaluated in TempEval-3 or proposed in later works. All systems are either end-to-end systems that address both the subtasks of temporal expression recognition(TER) and normalization, or partial systems that address only one of the subtasks. An overview of all presented existing systems can be found in Table 2.2.

Rule-based systems that rely only on hand-crafted rules have been applied with great success to temporal tagging. However, data-driven systems that use machine learning techniques to learn parameters given training data are becoming increasingly competitive. In the prediction of the value attribute, which identifies the time referred to in an expression, hybrid systems that employ a combination of learned parameters and predefined rules are considered state of the art.

In the subtask of TER, purely data-driven systems are slowly starting to catch up, but are still inferior to the best rule-based models. Fully data-driven normalization systems for TIMEX3 are currently not feasible with the available methods, since the task requires predicting a fine-grained and precise string from a huge output space detailing the exact time.

Table 2.2 Properties of existing systems for temporal tagging. FT is the unaligned model with FastText representations, BT is the unaligned model with BERT representations (Lange et al., 2020). It can be seen that most existing systems normalize to the TIMEX3 schema.

	Method		Scope		Annotation
	Rule-based	Data-driven	Recognition	Normalization	Schema
HeidelTime	✓		✓	✓	TIMEX3
SUTime	✓		✓	✓	TIMEX3
SynTime	✓		✓		TIMEX3
Timen	✓			✓	TIMEX3
NorMA	✓			✓	TIMEX3
UWTime	✓	✓	✓	✓	TIMEX3
ManTIME	✓	✓	✓		TIMEX3
Chrono	✓	✓	✓	✓	SCATE
Timenorm	✓	✓	✓	✓	SCATE
ClearTK		✓	✓		TIMEX3
ANNTIME		✓	✓		TIMEX3
Lange et al. (FT)		✓	✓		TIMEX3
Lange et al. (BT)		✓	✓		TIMEX3

That means that any system that focuses only on recognition must currently be accompanied by a rule-based normalization system in order to solve the full task of end-to-end temporal tagging.

Rule-based Systems

HeidelTime (Strötgen and Gertz, 2010) is a regular expression-based end-to-end system that has been extended multiple times over the years to include an extensive rule set for more and more languages and domains. In its most basic form, a rule in HeidelTime is a triple of a regular expression used for recognition, an associated normalization function that outputs the value attribute, and an associated type. These rules operate on tokens with associated POS-tags that are grouped into sentences, so input texts are subjected to a preprocessing pipeline of sentence splitting, tokenization and POS-tagging. HeidelTime can also be tuned to match a new data set, domain or language by adjusting the rules based on positive and negative examples. However, this cannot be considered a data-driven approach, since it always requires some manual corrections (Strötgen and Gertz, 2015).

SUTime (Chang and C. Manning, 2012) is a regular expression-based end-to-end system that is integrated in the Stanford CoreNLP pipeline (C. D. Manning et al., 2014) and built on top of its TOKENSREGEX framework. SUTime iteratively builds internal temporal representations by mapping tokens to temporal representations and temporal representations to compositional temporal representation. Filtering rules are employed to remove tokens that likely do not belong to a temporal expression. Like HeidelTime, SUTime operates on POS-tagged tokens.

SynTime (Zhong, Sun, and Cambria, 2017) is a TER-only system that builds on the regular expressions provided by SUTime, but introduces an additional layer of abstraction by identifying relevant tokens with the syntactic types time token, numeral and modifier. Hand-crafted heuristic rules are used to further merge the identified time tokens into time segments and finally into time expressions. SynTime does not identify the type of temporal expressions, only the extent. Like HeidelTime, SynTime has a protocol for the expansion of the rule set given a text. However, since this mechanism has been shown to decrease performance on some data sets, this work will only consider the unextended version of SynTime.

Timen (Llorens et al., 2012) is a normalization-only system that uses an external knowledge base as well as an external rule base to normalize recognized temporal expressions. The knowledge bases were originally intended as a community-sourced universal resource for temporal expression normalization, but the effort has since been discontinued. It predicts only the value attribute of a given temporal expression and relies on the type for normalization.

NorMA (Filannino, 2012) is a regular expression-based normalization-only system that is based on the TempEval-2 system TRIOS (UzZaman and J. Allen, 2010). It predicts both type and value for a given temporal expression. However, its rules are both hand-crafted and hard-coded, which makes the value prediction brittle for texts outside of its intended domain.

Hybrid Systems

UWTime (Lee et al., 2014) is a partly data-driven end-to-end system that is based on Categorical Combinatorial Grammars (CCGs) and operates on a combination of hand-crafted and automatically generated lexicon entries. A CCG is defined by a lexicon, which maps words to abstract categories and a set of combinators, which in turn provide rules for how to combine the categories (Steedman, 1987). Some categories like those associated with verbs denote functions that can be applied both forward and backward. Features for each token are its context tokens, its POS, one of thirteen manually defined lexical groups as well as the existence of determiners.

ManTIME (Filannino, Brown, and Nenadic, 2013) is a TER-only system that is based on CRFs. The input features of the CRF are 94 hand-crafted morphological and syntactic features as well as features derived from the ontology WordNet (Miller et al., 1990). They also draw on knowledge from predefined collections of instances of a common word type, so-called gazetteers. The output of the CRF are probabilities over BIO tag sequences, which are postprocessed with a set of empirically derived thresholding rules. In TempEval-3, the resulting expressions were normalized with the rule-based normalization-only system NorMA.

Chrono (Olex et al., 2018) is a mostly regular expression-based end-to-end system that normalizes to the SCATE annotation schema rather than TIMEX3, which means that it does not predict the type and value of a temporal expression, but rather constructs a compositional temporal entity for each temporal expression (Bethard and Parker, 2016). Features for the rule-based recognition are the tokens, their POS tags and whether they are numeric or temporal tokens as determined with hand-crafted regular expressions. Normalization is also largely rule-based, but uses SVMs, Naive Bayes Classifiers, Feedforward Neural Networks and Decision Trees to distinguish between periods and calendar intervals based on boolean features derived from context words.

Timenorm (Laparra, Xu, and Bethard, 2018) is a partly neural end-to-end system that, like Chrono, predicts SCATE entities rather than TIMEX3 tags. The authors propose two models based on character-level and word-level GRUs with different activation functions. Next to the tokens themselves their Unicode categories and POS tags are utilized. For normalization to SCATE entities, a rule-based system is used. In a further extension of Timenorm (Xu,

Laparra, and Bethard, 2019), pretrained character-level FLAIR embeddings are used as input features, which leads to a significant increase in performance.

Data-driven Systems

ClearTK (Bethard, 2013) is a TER-only system that uses CRFs, support-vector machines and logistic regression to predict BIO labels. The input features are a token's stem, POS tag, Unicode categories as well as additional features derived from a small set of time-word gazetteers. In TempEval-3, the resulting expressions were normalized with the rule-based normalization-only system Timen.

A popular neural TER-only architecture is the combination of LSTMs and CRFs with BIO-style output labels. This architecture is inspired by successful systems for the related task of NER (Lample et al., 2016). ANNTIME (Etcheverry and Wonsever, 2017) use LSTMs with GloVe word embeddings as input features and BILOU output format. More recently, Lange et al. use a neural LSTM-CRF architecture and experiment both with static FastText embeddings and contextualized BERT word representations as input features (Lange et al., 2020). In order to produce a multilingual model, they additionally experiment with an embedding alignment step and jointly train their system on multiple languages.

2.3 Knowledge Augmentation

Supervised machine learning can only learn to effectively address tasks for which there is a sufficient number of labeled examples available. While the amount of annotated text for temporal tagging in general is small compared to other tasks, this issue becomes really problematic at the token level. Named temporal expressions such as rare holidays may by chance never appear in the training data, but occur in the test data. In fact, it is likely that many named temporal expressions never occur in train data sets of the size that is currently used (Brucato et al., 2013). In addition to that, text phrases such as event names that only refer to a unique time given their context and external knowledge can sometimes only be interpreted correctly with background knowledge (Kuzey, Setty, et al., 2016).

One approach to alleviating this issue is incorporating external knowledge, which could contain e.g. information about all holidays. Many existing temporal tagging systems already have included this kind of knowledge, either explicitly as additional features in the case of data-driven models (ManTIME, ClearTK) or implicitly as part of hand-crafted rules (HeidelTime, SynTime). Augmenting neural systems with external knowledge has proven to be a successful strategy for many NLP tasks (K M, Basu Roy Chowdhury, and Dukkupati,

2018), such as coreference resolution (H. Zhang et al., 2019), literature retrieval (Zhao et al., 2019) and named entity recognition (Seyler et al., 2018).

Knowledge augmentation for neural models usually involves a retrieval step from an external knowledge source such as a knowledge graph or another form of knowledge base such as sets.

Graph-based Knowledge

In the case of graph-based knowledge, a context-based embedding method like the one proposed in Word2vec by Mikolov et al. is used to represent nodes or relations between nodes as real-valued vector representations (K M, Basu Roy Chowdhury, and Dukkipati, 2018). Another method to effectively integrate graph-based knowledge in neural models are the contextualized word representations derived from the pretrained knowledge-augmented neural language model KnowBERT (M. E. Peters et al., 2019). These word representations are enriched with graph-based knowledge retrieved from WordNet (Miller, 1995) as well as Word2vec-like embeddings derived from co-occurrence statistics of the titles and descriptions of pages in the online encyclopedia Wikipedia (Ganea and Hofmann, 2017). The resulting knowledge-augmented BERT representations have been shown to lead to improvements in many natural language understanding tasks including the sequence labeling task NER (M. E. Peters et al., 2019).

Set-based Knowledge

For knowledge bases that are set-based rather than graph-based, vectors can be derived e.g. by learning representations from boolean indicators of occurrence of a given sequence in the knowledge base. One common set-based knowledge source in natural language understanding systems are gazetteers, collections of instances of a common type. Set-based knowledge can also be used to compute statistics that are directly introduced as numerical features, e.g. the probability of a token being linked to any named entity Wikipedia page for the task of NER. (Seyler et al., 2018). This kind of knowledge would not be considered graph-based, since it does not really exploit the graph structure in the hyperlinks but only the boolean indication of a token being part of a link.

Chapter 3

Resources

In order to develop and evaluate a supervised knowledge-augmented system, at least three resources are necessary: A train data set to learn the parameters of the system, a test data set to evaluate the performance of the system, and a knowledge source from which auxiliary information about related to the examples in the train and test data sets can be retrieved. The following will give an overview of the data sets and knowledge sources that were used in this work.

3.1 Data Sets

In the experiments, all systems are evaluated on TIMEX3 data sets from the news and voice assistant commands. The TIMEX3 schema is chosen because most existing data sets are encoded in TIMEX3 and most existing systems normalize to it. In order to do the kind of comprehensive evaluation and comparison with existing systems that is the objective of this work, it is therefore necessary to use the predominant TIMEX3 schema.

The news domain is selected because of its status as the de-facto default domain of temporal tagging, as illustrated by the large number of richly annotated data sets and highly performant systems for temporal tagging on news data. Precisely due to the fact that most existing systems were created with the news domain in mind, it is interesting to evaluate them on a domain that differs from news in many core properties, voice assistant commands. When talking to a personal assistant, the sentences are on average shorter, the vocabulary is simpler and temporal expressions more often refer to the future rather than to the past (Zarcone, Alam, and Kolagar, 2020). Furthermore, successful knowledge augmentation for personal assistants would open up the possibility of personalized knowledge bases that ensure that personalized events can be successfully recognized and normalized, e.g. by using

a personal calendar as a knowledge base. Aggregated descriptive statistics about the two data sets that were used and their differences can be found in Table 3.1.

News Domain

For the news domain, the TempEval-3 (UzZaman, Llorens, et al., 2013) data set is chosen, since it is by far the largest news data set available at 768k tokens and is well-established as a benchmark data set for temporal tagging. The train data consists of 95k human-annotated tokens from TimeBank and AQUAINT. In addition to that, there is TempEval-3 Silver, which contains 666k words from the newswire corpus Gigaword (Napoles, Gormley, and Durme, 2012) that were annotated automatically with a weighted ensemble of temporal tagging systems that were successful in previous competitions. Previous works (Bethard, 2013) have cast doubt on the quality of both the AQUAINT portion and the machine-generated Silver portion of the TempEval-3 train data, with some authors going as far as manually fixing the mistakes (Lee et al., 2014). However, its large size makes the TempEval-3 data set well-suited to machine learning methods that require a large number of training examples such as neural networks. All systems are evaluated against the annotated ground truth in the expert-annotated test data set TempEval-3 Platinum with 6k tokens.

Voice Assistant Commands Domain

For the voice assistant commands domain, the PÂTÉ (Zarcone, Alam, and Kolagar, 2020) data set is selected. Apart from being a completely different domain, it is minuscule in size compared to TempEval-3, which means the systems are evaluated in a low-resource setting. Like TempEval-3, the annotations for PÂTÉ are produced by a human expert. Four out of 480 commands had to be discarded, since their unusual syntactic structure led to irreparable

Table 3.1 Descriptive statistics of used temporal tagging data sets. Total numbers were estimated from the annotated data set when not specified by the authors (UzZaman, Llorens, et al., 2013; Zarcone, Alam, and Kolagar, 2020). It can be seen that TempEval-3 and PÂTÉ differ in many aspects, the most prominent of which are absolute size, density of temporal expressions and average sentence length.

	Total Number			Average per Sentence	
	Sentences	Temp. Expressions	Tokens	Temp. Expressions	Tokens
TempEval-3	38 752	17 427	768 075	0.45	19.82
PÂTÉ	499	767	5 633	1.54	11.29

errors in some of the rule-based systems. Empty content tags, which only have the attributes value and type but no extent, were likewise removed because they are not supported by most of the systems. Due to the small size of PÂTÉ and the absence of a canonical train-test split, the data set is split into seven complementary chunks of 68 commands. The commands for each split are sampled randomly without replacement and the splits are kept fixed for all experiments. Predictions for each chunk are made with systems that are trained only on the other six chunks, effectively implementing a sevenfold crossvalidation procedure. HeidelTime has a specific setting intended for use with colloquial corpora, however, since preliminary experiments showed that it performed worse on PÂTÉ than the default setting, it was not used in the experiments.

3.2 Knowledge Sources

While a great wealth of human-curated structured knowledge is available in freely available knowledge graphs such as YAGO (Suchanek, Kasneci, and Weikum, 2007) and DBPedia (Auer et al., 2007), most of it is entity-centric and is not consistently annotated with temporal information (Gottschalk and Demidova, 2019). However, many existing temporal tagging systems, both rule-based and data-driven, already rely on human knowledge. In many cases, this knowledge is hard-coded in rules and difficult to extract, but some developers choose to intentionally expose their knowledge base in order to make their systems more accessible and modifiable. These exposed knowledge bases are often stored in the form of set-like collections. The following provides an overview of both the set-based knowledge sources and the graph-based knowledge sources that were used in this work.

Set-based Knowledge

Over the years, numerous temporal tagging systems have been proposed. Most of them include hand-engineered rules that are meant to encode the kind of knowledge that is necessary for temporal expression processing. However, some of the larger rule-based systems store their knowledge in separate folders in order to avoid rule duplication. Since attempts to centralize these results did not catch on, each system has built from the ground up its own set of resources that encode this kind of knowledge, and covers a slightly different share of the knowledge necessary to recognize and normalize temporal expressions (Llorens et al., 2012).

The most often exposed form of knowledge base is gazetteers. They are used to generate features for the data-driven systems ManTIME and ClearTK, and also in the form of

collections of regular expressions in the rule-based system HeidelbergTime. They are publicly available and are by design meant to encode mostly temporal knowledge, although some systems also include general world knowledge in their gazetteer as examples of expressions that are not temporal.

HeidelbergTime is perhaps the most comprehensive temporal tagging system, with support and dedicated resources for over 200 languages (Strötgen and Gertz, 2015). The developers explicitly expose their language-dependent resources¹ in the form of regular expressions that are grouped into 78 files by semantic context. There are sets of named temporal events such as "during the 10th National Hockey League All-Star Game" as well as sets of time tokens such as names of months or holidays. It also has semantic groups that indicate temporal ordering, such as the words "next" or "last". As all HeidelbergTime resources are time-related, it can be regarded as a set-based source of purely temporal knowledge.

The developers of ManTIME (Filannino, Brown, and Nenadic, 2013) make their custom gazetteers² available online as well. Like HeidelbergTime, these gazetteers are grouped into 13 files by content. Groups include time-related event names such as world festival names, but also many sets of tokens that are explicitly not time-related and that might enable a system to increase its accuracy by filtering out those non-temporal tokens at an early stage. Such negative-example gazetteer files are e.g. human names and named locations around the world. The ManTIME gazetteers interestingly also include a semantic group that contains the most common 200k words in the English language. They do not specify their intention, but perhaps these features can be used to identify rare words and names of unknown entities by exclusion. ManTIME resources are a set-based knowledge source of both general world knowledge and temporal knowledge.

In sum, HeidelbergTime resources are used as an example of a set-based temporal knowledge source, while ManTIME resources are used as an example of a set-based knowledge source that contains both general and temporal knowledge.

Graph-based Knowledge

Knowledge graphs can model a wide range of information about entities and their relations. There exist a number of large-scale projects to construct knowledge graphs that encode as much general or specific world knowledge as possible (Nickel et al., 2016). Popular freely available examples of such general knowledge graphs include DBPedia (Auer et al., 2007), Wikidata (Erxleben et al., 2014) and YAGO (Suchanek, Kasneci, and Weikum, 2007), which is an attempt to unify information from Wikipedia, WordNet and other sources. WordNet

¹<https://github.com/HeidelbergTime/heideltime/tree/master/resources/english/repattern>

²<https://github.com/filannim/ManTIME/tree/master/mantime/data/gazetteer>

(Miller, 1995) is a graph-like lexical knowledge base that encodes information about word senses. Although it is not their main focus, these general knowledge graphs in some cases also encode temporal knowledge about events. Furthermore, explicitly non-temporal information could also be useful for temporal tagging, since it can theoretically enable systems to ignore non-temporal tokens.

There also exist a few knowledge graphs that are specifically tailored to temporal tasks. The EventKG (Gottschalk and Demidova, 2019) builds on the previously mentioned general KGs, but focusses specifically on the collection and distillation of knowledge about events and temporal relations. Similarly, the Temporal Knowledge Base (Lacroix, Obozinski, and Usunier, 2020) is a subset of Wikidata that focuses on triples that contain timestamps that indicate a temporal range for which a given relation is valid. However, these temporal knowledge graphs mostly encode knowledge about events, which means they are not likely to be useful in domains that reflect day-to-day topics rather than political or historical events.

The neural language model KnowBERT (M. E. Peters et al., 2019) has been augmented both with graph-based knowledge from WordNet and with entity-centric knowledge about Wikipedia entities. The pretrained model made available by M. E. Peters et al. is used for this purpose. Since KnowBERT’s weights are not finetuned any further, the knowledge injected into the representations preserved and the representations can be considered a source of purely general knowledge.

In sum, KnowBERT representations are used as an example of a graph-based knowledge source that contains mostly general knowledge which is encoded in Wikipedia descriptions or in the graph structure of WordNet.

Chapter 4

Methodology

Since no methods for neural temporal expression normalization have been proposed yet, this work focuses on Neural TER (NTER) systems and techniques to extend them to Knowledge-Augmented NTER (KANTER) systems. The NTER systems can be combined with a rule-based normalization system to form a complete end-to-end temporal tagging system. In the following, a baseline NTER system will be presented as well a KANTER system that can integrate external knowledge from set-based and graph-based sources.

4.1 Neural Temporal Expression Recognition

Similar to previous work on neural methods for temporal tagging (Etcheverry and Wonsever, 2017), the proposed NTER architecture uses LSTMs as the main sequence processing module. In line with the unaligned model of Lange et al., the predicted labels are contextualized with a CRF as a top layer, yielding a LSTM-CRF architecture, which has already been applied with great success to many related sequence labeling tasks such as NER (Lample et al., 2016).

As input, the system accepts tokenized sentences. The tokens are embedded using pretrained word representations and fed into a LSTM-CRF architecture, which outputs labels from the set of all meaningful combinations of BIO tags with each TYPE. Consequently, there are nine possible labels for each token:

- B-TIME, I-TIME
- B-DATE, I-DATE
- B-SET, I-SET
- B-DURATION, I-DURATION
- 0 (not a temporal expression)

A sequence of tokens labeled with these BIO labels can be transformed to TIMEX3-annotated text with a simple mechanism that takes into account "B-" labeled tokens as the start of an EXTENT, the last consecutive "I-" label as the end of the EXTENT and infers the TYPE from the label type. The result is a text with TIMEX3 tags that have an EXTENT and TYPE, but no VALUE, thereby fulfilling the task of TER. The overall structure of the NTER architecture can be seen in Figure 4.1.

Word Representations

Pretrained general-purpose word vectors are used to bootstrap the input features of the system, since the task of TER typically lacks both the sufficient amount of data and a suitable objective to learn semantic word representations on the fly. Following previous work by Lange et al., both static and contextualized word representations are considered.

For static word representations, a real-valued neural embedding layer is initialized with pretrained word vectors for the two million most frequent words. The weights of this neural embedding are trained jointly with the system in order to focus the representations on temporal aspects of token meanings. The employed word vectors of dimensionality 300 are downloaded from the authors of Word2vec (Mikolov et al., 2013), GloVe (Pennington, Socher, and C. Manning, 2014) and FastText (Bojanowski et al., 2017), who induced them by training their models on corpora with vast amounts of web data such as Common Crawl¹.

The pretrained language model BERT (Devlin et al., 2019) is used to obtain contextualized word representations. Following the methodology Devlin et al. use for feature-based NER, BERT is not finetuned, but instead used as a frozen feature-extractor. BERT separates tokens into subtokens using WordPiece tokenization (Wu et al., 2016), which means it may generate multiple representations for a single word. The concatenation of the activations of the last four layers of BERT for the first sub-token is taken to be the representation of each token, as recommended by Devlin et al. This initially yields word vectors of dimensionality $4 \cdot 768 = 3072$, which is reduced to 300 using a learned FFNN to ensure a representational capability that is comparable to that of static word representations.

Inspired by the successes of Xu, Laparra, and Bethard with contextualized character embeddings, each character in each word can optionally also be embedded with learned randomly initialized vectors of dimensionality 50. The representations of the characters in a word are then concatenated and contextualized by applying a Convolutional Neural Network (CNN) with a window size of 3 to extract local character-level trigram features. The trigram representations for each word are aggregated with a maxpool operation over all characters

¹<https://commoncrawl.org/>

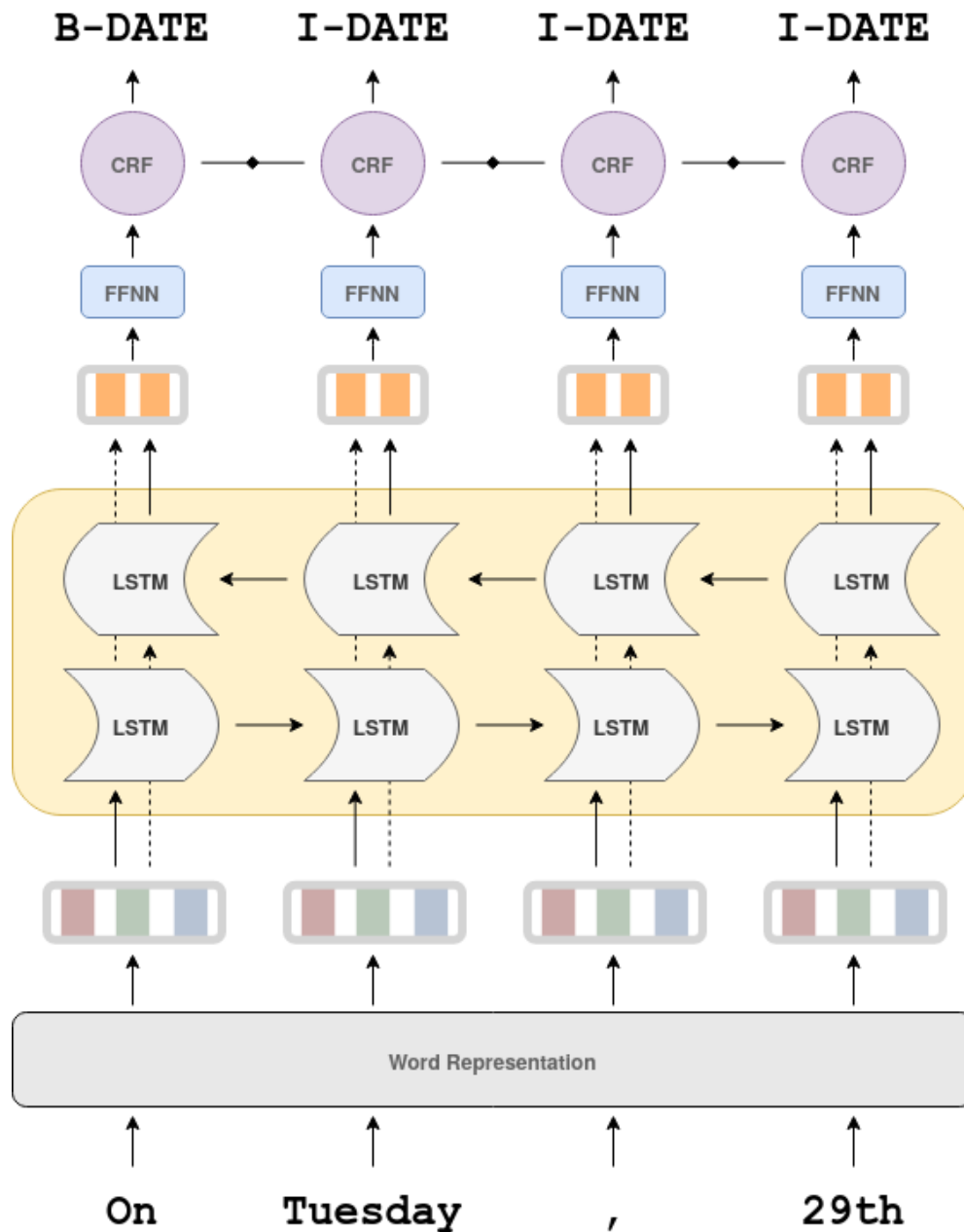


Fig. 4.1 Architecture of NTER system with example sentences. Each word in the sequence is first represented and then contextualized in the bidirectional LSTM. A single FFNN maps each resulting intermediate word representation to probabilities over all possible labels. The CRF contextualizes the intermediate predictions and the Viterbi algorithm yields the final predicted label sequence.

and concatenated with the word representations. A similar setup has been shown to perform well on the related tasks of NER and POS tagging (Ma and Hovy, 2016).

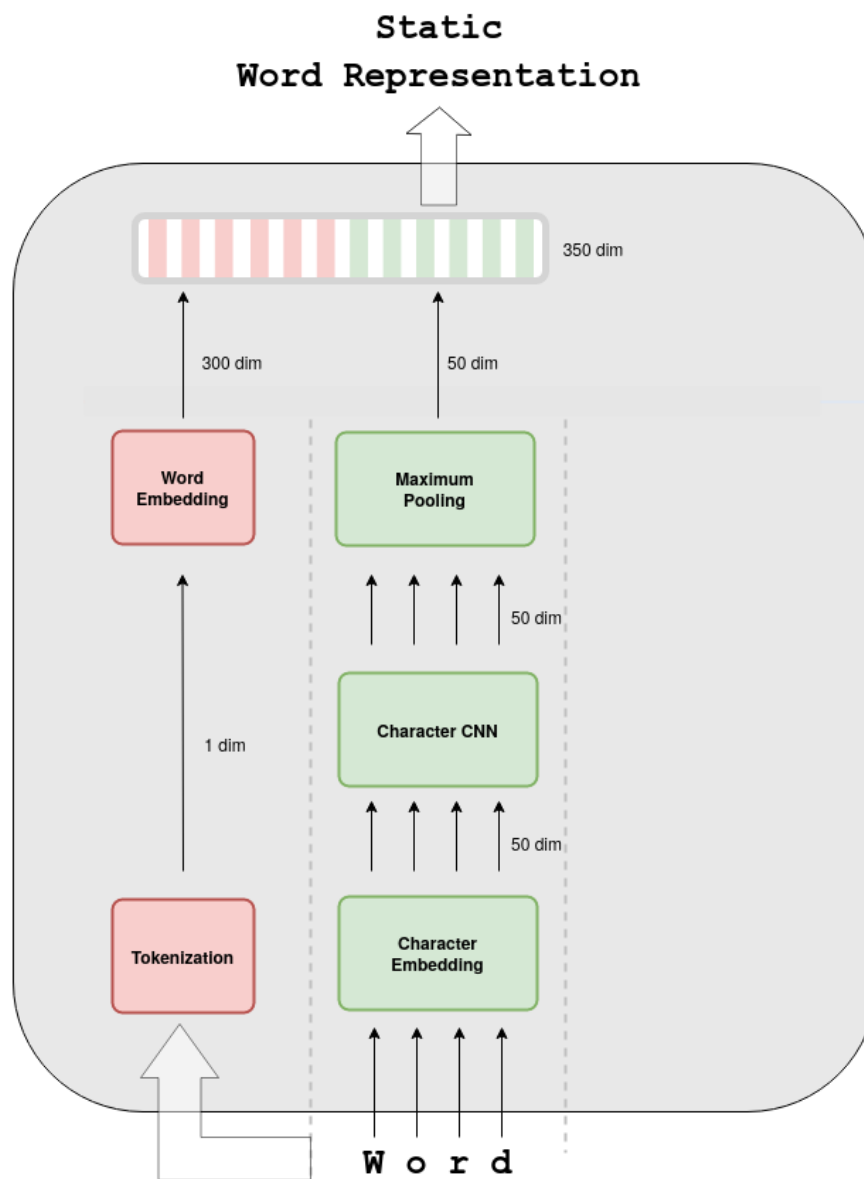


Fig. 4.2 NTER static word representation module including character embeddings.

A visualization of the architecture and dimensionality of the static and contextualized word representation modules in NTER can be found in Figures 4.2 and 4.3.

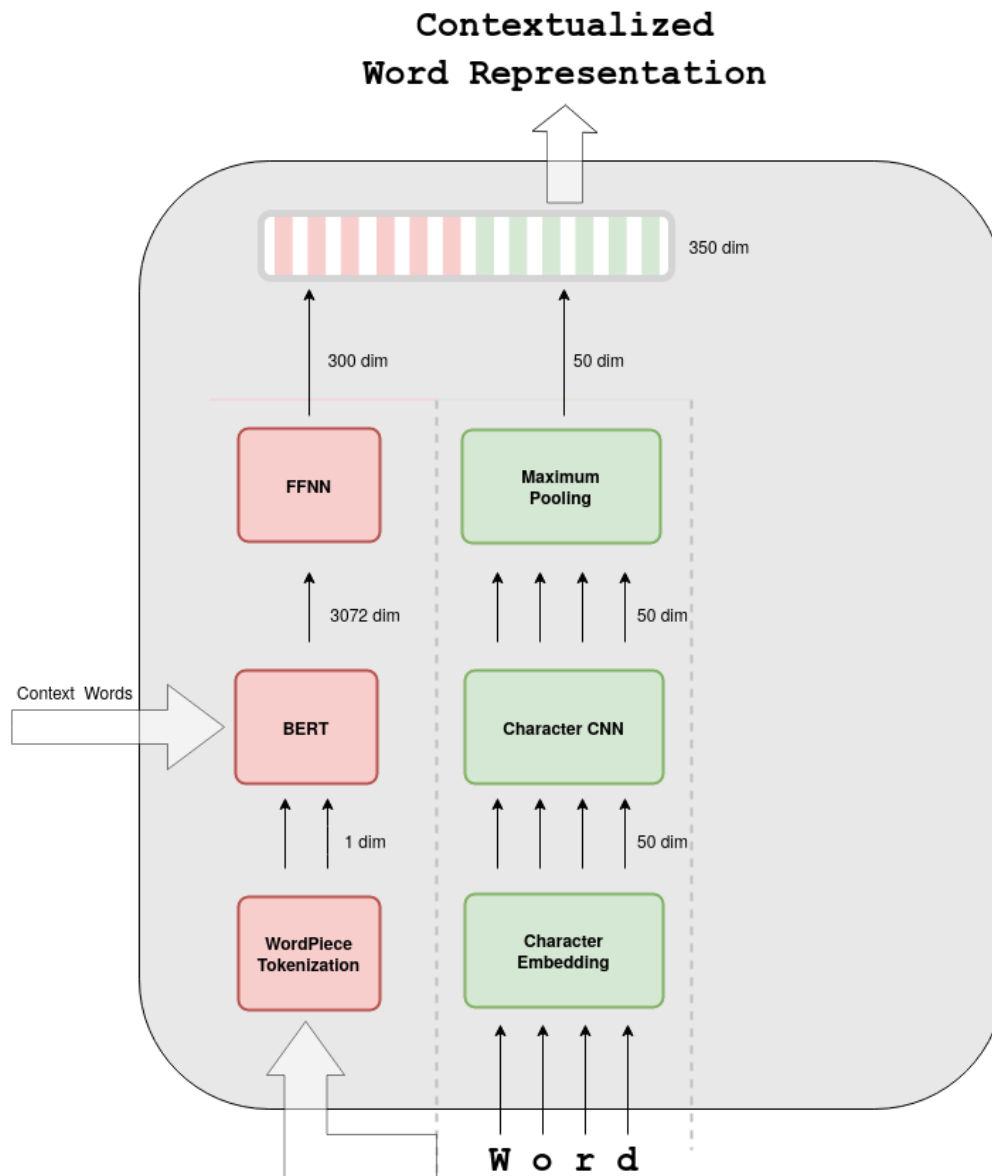


Fig. 4.3 NTER contextualized word representation module including character embeddings.

LSTM

The employed LSTM architecture follows closely the architecture described by Lange et al. It is bidirectional, which means there are two LSTM units that consume the sequence and whose hidden states for each word are concatenated to yield contextualized intermediate word representations. Since every representation should be exposed to information from both previous and later words, one unit consumes the sequence in the original order, while the

other consumes it in reverse order (Jurafsky and Martin, 2009). Each unit yields a hidden state of dimensionality 128, which leads to a dimensionality of $2 \cdot 128 = 256$ for the intermediate word representations. A dropout of 0.5 is applied to these representations in order to prevent overfitting, before a learned FFNN is used to predict intermediate probabilities for each label from the intermediate token representations.

Conditional Random Field

The final label probabilities are obtained by contextualizing the intermediate predictions with a linear-chain CRF (Lafferty, McCallum, and Pereira, 2001). Linear-chain CRFs can be considered extensions of HMMs with weaker independence assumptions for the input space (Klinger and Tomanek, 2007), which is sensible in the case of a LSTM-CRF architecture given that the CRF input stems from representations that were exposed to context from both directions in the LSTM stage. Nevertheless, further contextualization of the label predictions has been shown to perform better than simply normalizing the intermediate predictions with softmax (Yang, Liang, and Y. Zhang, 2018). This is also intuitive for the BIO label set, since it encompasses a grammar-like ordering that makes some subsequences of labels illegal, such as an "I-" label without a preceding "B-" label (Lample et al., 2016).

A CRF computes the score for a sequence of labels given a sequence of elements by multiplying arbitrary weighted feature functions that operate on the previous label, the current input and the current position for each element in the sequence (Klinger and Tomanek, 2007). In practice, in the LSTM-CRF architecture this is implemented through addition in exponential space of values from a real-valued transition score matrix to the input features. The transition score matrix contains bigram compatibilities between labels and is learned during training, so one would expect the bigram compatibility of e.g. the label subsequence "O", "I-DATE" to be low, while "B-DATE", "I-DATE" would be high.

The CRF is trained by maximizing the log-probability of the correct label sequence, which at the same time minimizes the log-probability of incorrect label sequences, since sequence probabilities are normalized via softmax (Lample et al., 2016). During inference, the Viterbi algorithm is used to yield the most probable label sequence given the intermediate predictions of the LSTM (Bishop, 2006).

4.2 Knowledge-augmented Neural Temporal Expression Recognition

In order to make the systems as comparable as possible, the KANTER system employs the exact same methodology and parameters as the baseline NTER system and only modifies the token-level representations by injecting external knowledge. In that way, the KANTER system can retain the strengths of the baseline system and learn to only draw on the knowledge in cases where it is useful.

Set-based Knowledge

Given set-based knowledge in the form of semantically grouped gazetteers, they are converted into disjunctive regular expressions. The individual expressions may be single words, such as the time-token "hour", or multi-word expressions such as the holiday "Assumption of the Blessed Virgin Mary into Heaven". These disjunctive regular expressions are applied to the input sentences in order to yield token-level boolean features that indicate whether a token belongs to an expression in a semantical group or not.

The boolean membership features are reduced to a single categorical feature by assigning a null label if the token is part of no semantical group and otherwise selecting the most relevant matching semantical group according to a predefined hierarchy. This hierarchy is computed based on the number of words in the regular expression, with the longest regular expressions having the highest priority. The intuition behind this is that expressions with more words are generally more specific than expressions with fewer words and should therefore supersede them.

Motivating examples are the phrases "4th of July" and "Easter island". As a holiday, the phrase "4th of July" has a special meaning that goes beyond what is captured in the time tokens "4th" and "July", although this would likely not yield to an error in a recognition system, since both are temporal expressions. While "Easter" on its own usually indicates a time, in the context of the location in the Pacific Ocean "Easter island" however, it does not. Therefore, if the token "Easter" is flagged as a time token, it is likely to lead to misclassification. Given that the information that "Easter island" belongs to the semantic group of locations is available, it should thus supersede the information that "Easter" is a time token.

A similar phenomenon of supersedence of temporal scopes was noted in previous literature (Kuzey, Strötgen, et al., 2016). The practice of selecting the longest match in a gazetteer as a categorical feature has also been employed in previous work (Seyler et al.,

2018), although not with this explicit motivation. Looking for shorter expressions only after the computationally less expensive search for larger strings has not yielded a match is also reminiscent of cascading classifiers for object detection, which consider more expensive classifiers only after lightweight classifiers have not come to a definitive conclusion (Viola and Jones, 2001).

The resulting categorical feature is represented numerically using a randomly initialized neural embedding of dimensionality 50, whose weights are learned during training. These knowledge category embeddings are meant to represent both the information that the token belongs to a certain semantic group and that the fact that the token does not belong to any of the semantic groups with a higher priority. Compared to representation as one-hot encoded vectors that would encode the same information, the projection into a space with 50 dimensions represents a dimensionality reduction for the 78 semantic groups in HeidelbergTime, which is expected to encourage the filtering of information that is not useful. For the 13 ManTIME groups, an embedding in 50 dimensions correspond to a dimensionality expansion, however in ManTIME there are both groups that are expected to correlate positively with temporal expressions, such as names of festivities, and groups that are expected to correlate negatively with temporal expressions, such as names of countries or persons. As a consequence, it is a sensible precaution to provide more representational space than strictly necessary for ManTIME categories, since positively and negatively correlated features might have to be stored in a disentangled (Liao et al., 2020) space in order to be useful. The knowledge embedding dimensionality is also intentionally kept fixed across different knowledge sources in order to enable comparisons of the content of the knowledge sources without dimensionality as an additional influence.

Like character embeddings, the knowledge embeddings are concatenated to the pretrained word representations before feeding them into the LSTM-CRF. This setup is intentionally kept as general as possible to allow for the integration of arbitrary knowledge-derived categorical features. A visualization of the architecture and dimensionality of the knowledge-augmented static word representation module in KANTER can be found in Figure 4.4

Graph-based Knowledge

Given graph-based knowledge in the form of triples, a method is needed to select the relevant node in the graph and to embed the discrete information about the selected node and its relations as real-valued vectors for use in the neural system. Previous work has shown that the effective training of both the retrieval and embedding mechanism requires more annotated data than available for temporal tagging and ideally supervision that is directly related to

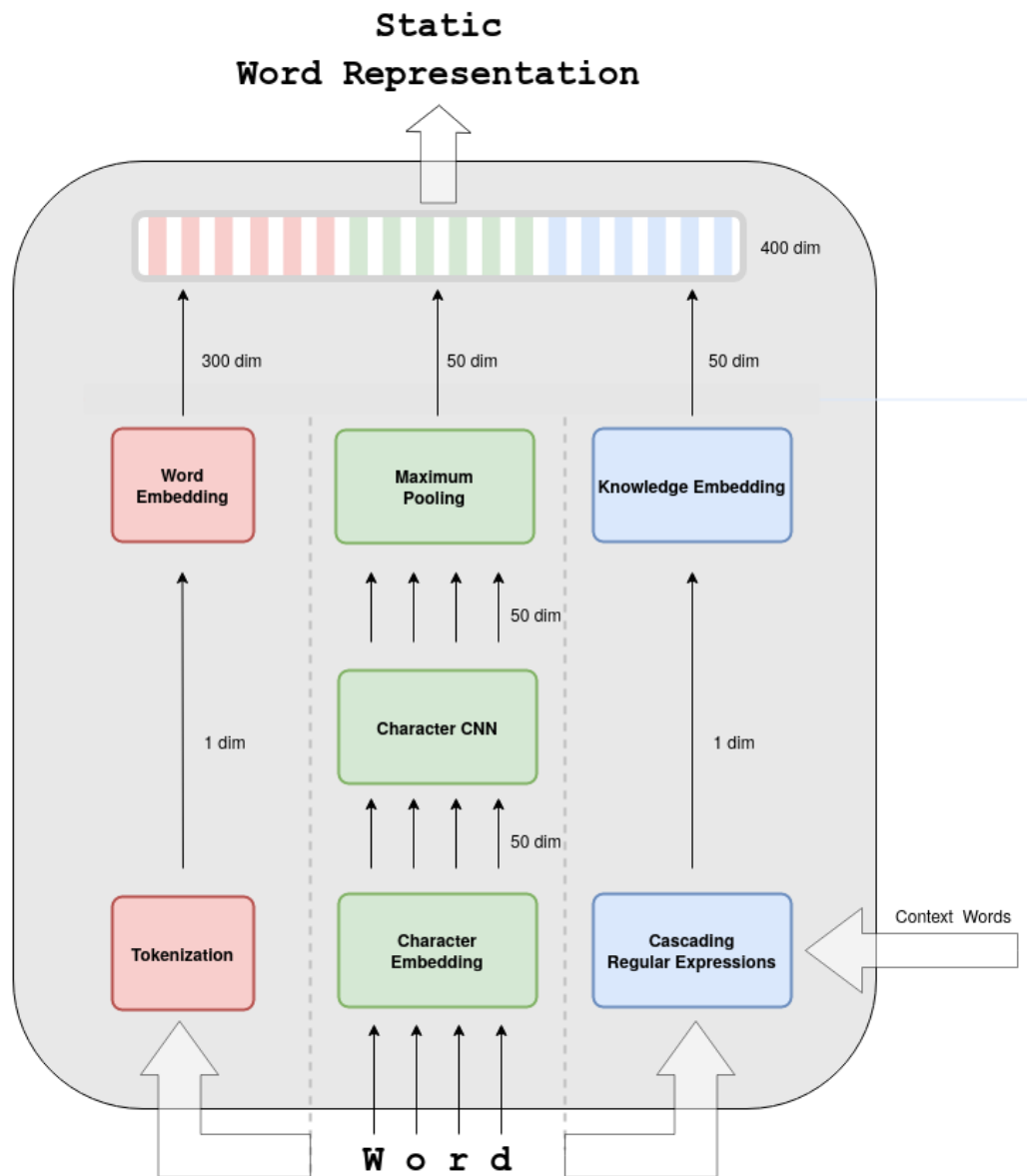


Fig. 4.4 KANTER contextualized word representation module including character and knowledge embeddings.

the entity-based knowledge in the graph to ensure a robust learning signal for retrieval and embedding (K M, Basu Roy Chowdhury, and Dukkupati, 2018).

Therefore, the KANTER system uses the pretrained knowledge-augmented language model KnowBERT (M. E. Peters et al., 2019) instead of training its own graph retrieval and

embedding mechanism. KnowBERT is a version of BERT that has been pretrained on an entity linking task and on a word sense disambiguation task in order to encourage the model to incorporate injected information from WordNet (Miller, 1995) and the online encyclopedia Wikipedia.

For the graph-based WordNet, TuckER relation embeddings (Balazevic, C. Allen, and Hospedales, 2019) are used to induce embeddings over the relations of a given set of synonyms, while sentence embeddings (Subramanian et al., 2018) are used to encode its gloss, which is a brief definition of a word sense. The resulting vectors are concatenated and injected into BERT before the last layer. For Wikipedia, the page titles and their description texts are used to create entity embeddings with a training objective similar to that of Mikolov et al. in Word2vec (Ganea and Hofmann, 2017). The Wikipedia information is injected into BERT before the penultimate layer. Given that the knowledge is injected only in the last few layers, M. E. Peters et al. use only the activations of the last layer as contextualized token representations, yielding real-valued vectors of dimensionality 768 (M. E. Peters et al., 2019). As with BERT, these representations are reduced to dimensionality 300 in KANTER using a learned FFNN to ensure comparable representational capabilities.

4.3 Evaluation

The evaluation metrics defined TempEval-3 are widely considered standard for the evaluation of temporal tagging. When evaluating the extent of temporal expressions, a distinction is made between strict matches, where the predicted character span matches the annotated character span exactly, and relaxed matches, where there is only a partial overlap between the spans. For the evaluation of the attributes type and value, all relaxed matches between annotations and predictions are compared (UzZaman, Llorens, et al., 2013).

An instance of predictions and annotations of TIMEX3 tags can be found in Example 4.1. The extent attribute is implied by the position of the open and closing tags, while the attributes type and value are written inside the opening tag. Example 4.1 also highlights some of the challenges that TER systems face, such as the problem that the falsely predicted extent on its own also constitutes a valid temporal expression.

Example 4.1 Prediction (P) and ground truth (T) for TIMEX3 annotations of a temporal phrase.

P: It was <TIMEX3 type="DATE" value="P2Y"> two years </TIMEX3> ago
T: It was <TIMEX3 type="DATE" value="2011"> two years ago </TIMEX3>

The instance in Example 4.1 would be evaluated as follows by the different metrics:

- Strict Match EXTENT: Prediction false, since tags imply different character spans
- Relaxed Match EXTENT: Prediction true, since tags imply overlapping character spans
- TYPE: Prediction true, since relaxed match EXTENT and TYPE are equal
- VALUE: Prediction false, since relaxed match EXTENT but VALUE is different

The TempEval-3 evaluation protocol aggregates the boolean comparison results for all annotated and predicted temporal expressions into a F1 score, which is calculated as the harmonic mean of precision and recall:

$$f_1 = \frac{2 \cdot \textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}$$

For the extent, precision and recall are defined as follows:

$$\textit{precision}_{\textit{extent}} = \frac{|\textit{pred} \cap_{\textit{match}} \textit{annot}|}{|\textit{pred}|}$$

$$\textit{recall}_{\textit{extent}} = \frac{|\textit{pred} \cap_{\textit{match}} \textit{annot}|}{|\textit{annot}|}$$

Since the type and value attributes are only compared for those predictions with relaxed matching extent, a conditional term is added to the precision and recall formulas for those attributes:

$$\textit{precision}_{\textit{attribute}} = \frac{|\{x \in (\textit{pred} \cap_{\textit{match}} \textit{annot}) \mid \textit{if } x_{\textit{attribute}}^{\textit{pred}} = x_{\textit{attribute}}^{\textit{annot}}\}|}{|\textit{pred}|}$$

$$\textit{recall}_{\textit{attribute}} = \frac{|\{x \in (\textit{pred} \cap_{\textit{match}} \textit{annot}) \mid \textit{if } x_{\textit{attribute}}^{\textit{pred}} = x_{\textit{attribute}}^{\textit{annot}}\}|}{|\textit{annot}|}$$

For the F1 score of these attributes, the conditional term means that the attribute F1 score is essentially equivalent to the product of extent F1 score and attribute accuracy (UzZaman, Llorens, et al., 2013).

Relevant metrics for TER are strict match extent F1, relaxed match extent F1 and relaxed match type F1. The overall task of end-to-end temporal tagging is additionally evaluated with relaxed match value F1 score, which is the result of comparing the normalized value attribute to the ground truth as annotated in the data set.

Chapter 5

Experimental Setup

The following describes experiments that aim to evaluate the proposed NTER and KANTER methods, replicate the reported results of existing temporal tagging baseline systems and compare all systems on the task of end-to-end temporal tagging in the domains of news and voice assistant commands. All systems are evaluated both on the canonical train-test split of the large TempEval-3 news data set and with crossvalidation on the seven splits of the smaller PÂTE voice assistant commands data set. The following describes the evaluated systems and the used configurations in detail.

5.1 Systems

Since the used data sets are annotated with the TIMEX3 schema and the proposed NTER and KANTER systems normalize to the TIMEX3 schema, the choice of existing temporal tagging systems to evaluate is restricted to TIMEX3 systems. In Table 5.1, an overview of the reported performances of the existing systems on TempEval-3 test data can be found. Systems are selected from this pool with the goal of both comparing NTER and KANTER against the most successful systems as well as at least one system from each type of approach.

Based on performance, the rule-based TER-only system SynTime (Zhong, Sun, and Cambria, 2017) and the CCG-based end-to-end system UWTime (Lee et al., 2014) are chosen, since they are each considered state of the art on TempEval-3 test data with regards to two relevant metrics. HeidelTime (Strötgen and Gertz, 2010) serves as a representative of purely rule-based temporal tagging systems, due to its robust performance and extensive coverage of different domains and languages. Furthermore, ClearTK (Bethard, 2013) is selected to represent traditional non-neural machine learning methods such as support vector machines and logistic regression. The neural TER systems ANNTime and those presented by Lange et al. are not publicly available, so the proposed NTER method is chosen to represent

the existing neural methods in the experiments. This is reasonable, since the architecture of the proposed NTER system is very similar to existing neural methods in that it also relies on pretrained word representations, LSTMs and CRFs (Lange et al., 2020).

For existing systems, the publicly available implementations provided by the authors are used with default hyperparameters. The NTER and KANTER systems are implemented using the established Open Source Sequence Labeling Toolkit¹ (Yang and Y. Zhang, 2018) based on PyTorch (Paszke et al., 2019) with default hyperparameters of the toolkit, which are adopted from a detailed survey of hyperparameter settings for sequence labeling tasks reported in previous work (Reimers and Gurevych, 2017a). Contextualized word representations are derived using the Transformers² (Wolf et al., 2020) library with the pretrained model bert-base-cased in the case of BERT and the authors' implementation³ with the pretrained model knowbert-w+w in the case of KnowBERT (M. E. Peters et al., 2019). The optimizer Adam (Kingma and Ba, 2014) is used to train the parameters over 50 epochs, with an initial learning rate of 0.0001 that is subject to continuous decay. Early stopping on validation data is not employed, since there is no designated validation data available and the comparatively low number of training epochs combined with the dropout rate of 0.5 should prevent overfitting.

¹<https://github.com/jiesutd/NCRFpp>

²<https://github.com/huggingface/transformers>

³<https://github.com/allenai/kb>

Table 5.1 Reported F1 scores of TIMEX3 systems on TempEval-3 test data with TempEval-3 evaluation protocol. FT is the unaligned model with FastText representations, BT is the unaligned model with BERT representations. Results marked with "-" were not reported by the authors. Best results for each metric are underlined.

	Extent		Type	Value
	Strict	Relaxed	Relaxed	Relaxed
HeidelTime	81.34	90.30	82.09	77.61
SUTime	79.57	90.32	80.29	67.38
SynTime	<u>92.09</u>	<u>94.96</u>	-	-
UWTime	83.10	91.40	<u>85.40</u>	<u>82.40</u>
ManTIME	74.33	89.66	77.39	68.97
ClearTK	82.71	90.23	84.20	64.66
ANNTIME	79.15	-	74.04	-
Lange et al. (FT)	68.36	79.14	72.13	-
Lange et al. (BT)	73.09	84.34	75.50	-

For the TER-only systems SynTime, ClearTK and (KA)NTER, the existing rule-based normalization system Timen⁴ (Llorens et al., 2012) is used on top to form complete end-to-end temporal tagging systems. In the case of SynTime, the regular-expression based NorMA⁵ (Filannino, 2012) is additionally employed in order to predict the type attribute, since Timen requires it and SynTime does not provide it.

To summarize, the following systems and implementations are evaluated:

- UWTime⁶
- SynTime⁷ + NorMA⁵ + Timen⁴
- HeidelTime⁸
- ClearTK⁹ + Timen⁴
- NTER¹² + Timen⁴
- KANTER¹²³ + Timen⁴

5.2 Experiments

First, all combinations of word representations and character embeddings in the baseline NTER system are evaluated. With the four presented word representation methods Word2vec, GloVe, FastText, and BERT and the two options of using character embeddings or not using character embeddings, this yields $4 \cdot 2 = 8$ NTER configurations. Numerical hyperparameters are not tuned, since the goal is not to find the best possible neural system, but to establish a working baseline NTER system in order to investigate the effects of knowledge augmentation.

After that, the KANTER system is evaluated by adding KnowBERT with and without character embeddings to the existing NTER configurations and combining each NTER configuration with either no knowledge embeddings, or those derived from ManTIME and HeidelTime gazetteers. This results in $(8 + 2) \cdot 3 = 30$ evaluated KANTER configurations. The subset of eight configurations that was already evaluated in the NTER experiments is not evaluated again.

Given one training run on TempEval-3 and seven training runs on the PÂTÉ splits, the total number of experiments needed to evaluate each configuration once is $(1 + 7) \cdot (8 + 22) = 240$.

⁴<https://github.com/leondz/timen>

⁵<https://github.com/filannim/timex-normaliser>

⁶<https://bitbucket.org/kentonl/uwtime/>

⁷<https://github.com/xszhong/syntime>

⁸<https://github.com/HeidelTime/heideltime>

⁹<https://github.com/ClearTK/cleartk>

Due to this already large number of experiments, each NTER and KANTER configuration is evaluated only once based on a single training run per data set or data set split.

Finally, the four existing temporal tagging systems are evaluated on TempEval-3 and the PÂTÉ splits, which leads to $(1 + 7) \cdot 4 = 32$ further experimental runs. The great size of the TempEval-3 train data set presents a disadvantage for existing systems that were designed with smaller data sets in mind. For instance, UWTime is unable to handle the high memory consumption of the full TempEval-3 training data set and can only be trained on the human-generated subsets TimeBank and AQUAINT and not on the large machine-generated subset TempEval-3 Silver. Since this observation is also consistent with the choices Lee et al. when training on TempEval-3 data, UWTime is trained only on the human-generated portion of the TempEval-3 training data set. Similarly, Bethard reports worse results when training on the full TempEval-3 train data set rather than only on the human-annotated subset. However, since their system ClearTK can handle the full amount of data and it is desirable for all systems to have been exposed to the same data to ensure comparability, ClearTK is nevertheless trained on the full TempEval-3 train data set.

The official TempEval-3 evaluation script¹⁰ is used to follow the TempEval-3 evaluation protocol and calculate the evaluation metrics. For TER, the metrics strict extent F1, relaxed extent F1 and relaxed type F1 are relevant, the most important metric for end-to-end temporal tagging is relaxed value F1. It should be noted that the end-to-end performance is highly dependent on the performance of the previous TER step, since the most easily recognizable expressions are often also the most easily normalizable expressions.

¹⁰https://github.com/naushadzaman/tempeval3_toolkit

Chapter 6

Results

In the following, results of experiments with the NTER and KANTER systems on the task of TER will be presented and put in context with reported scores of other neural TER systems where available. After that, results of end-to-end temporal tagging experiments with the best NTER and KANTER systems combined with the existing rule-based normalization system Timen will be compared with the results of experiments with existing rule-based and data-driven systems.

6.1 Neural Temporal Expression Recognition

As can be seen in the results on the TempEval-3 news data set in Table 6.1, pretrained contextualized BERT representations seem to be the most effective word representations for NTER in all metrics, which is consistent with previous literature both on temporal tagging (Lange et al., 2020) and other natural language processing tasks (Devlin et al., 2019). Among the static word representations, GloVe yields the best results for extent identification, which is consistent with the superior performance of ANNTIME over the systems proposed by Lange et al. as measured by strict match extent F1 score.

In absolute terms, the proposed NTER system overall seems to perform much better than previous neural TER systems with similar word representations. These differences can partly be explained by differences in the architectural choices and training protocols. ANNTIME uses only a LSTM architecture, not LSTM-CRF. Lange et al. employ a LSTM-CRF architecture, but train their English model jointly with other languages.

The extension with character embeddings generally seems to improve performance on extent identification for all word representations except for GloVe embeddings, for which it decreases performance significantly.

The experimental results on the voice assistant commands data set PÂTÉ, which can be found in Table 6.2, likewise indicate that BERT representations are the best choice for word representations, with GloVe embeddings again coming in second in terms of extent identification performance. On PÂTÉ, the positive effect of character embeddings is much more pronounced and consistent, increasing performance for GloVe embeddings as well. A quantitative analysis of all NTER word representation combinations shows that the use of character embeddings improves the relaxed match F1 score on average by 0.75 for TempEval-3 and by 0.86 for PÂTÉ.

6.2 Knowledge-augmented Neural Temporal Expression Recognition

As Figure 6.1 shows, augmenting the best word representations for NTER with external knowledge does not have a consistently positive effect on extent identification as measured by relaxed match F1 score. Combining GloVe representations with knowledge-derived features leads to significant increases or decreases in performance, depending on data set and knowledge source, further illustrating the volatility of GloVe embeddings that was observed

Table 6.1 Reported F1 scores of existing neural TER methods and F1 scores of proposed NTER system on TempEval-3 test data with TempEval-3 evaluation protocol. Best results for each metric are underlined.

	Representation		Extent		Type
	Word	Character	Strict	Relaxed	Relaxed
ANNTIME	GloVe		79.15	-	74.04
Lange et al.	FastText		68.36	79.14	72.13
Lange et al.	BERT		73.09	84.34	75.50
NTER	Word2vec		77.69	82.64	78.51
NTER	Word2vec	✓	79.34	83.47	76.86
NTER	GloVe		83.53	86.75	79.54
NTER	GloVe	✓	81.78	85.02	75.30
NTER	FastText		76.68	83.79	73.25
NTER	FastText	✓	78.57	84.13	74.60
NTER	BERT		82.07	<u>86.85</u>	79.68
NTER	BERT	✓	<u>84.00</u>	86.40	<u>80.80</u>

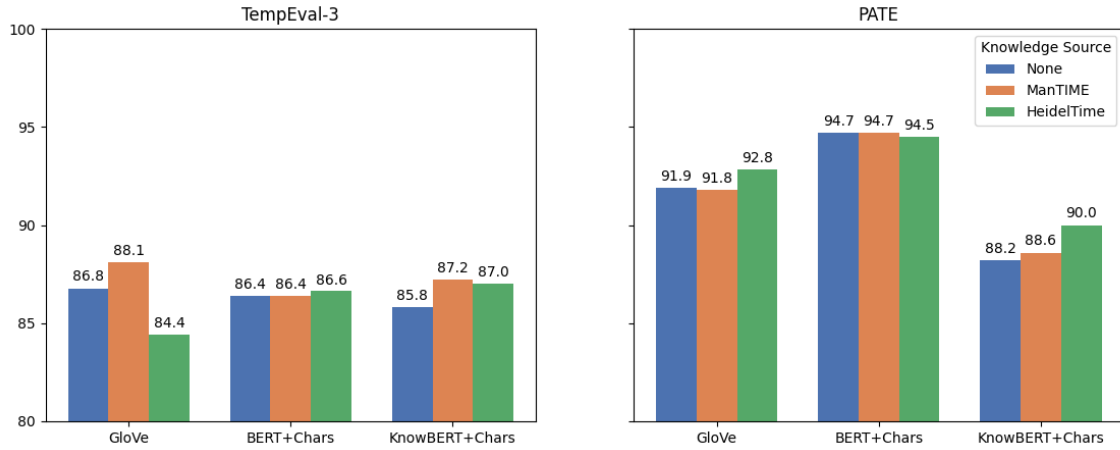


Fig. 6.1 Relaxed match F1 scores of proposed KANTER system with different knowledge sources on TempEval-3 test data and crossvalidation on PÂTÉ with TempEval-3 evaluation protocol.

in combination with character embeddings. Performance with BERT representations is largely unaffected by additional external knowledge. KnowBERT representations are on their own worse than BERT representations, but improve consistently when combined with other knowledge sources.

In order to investigate the effects of knowledge augmentation in general and not just on the best systems, a quantitative analysis of all NTER representation and knowledge source

Table 6.2 F1 scores of proposed NTER system on PÂTÉ crossvalidation with TempEval-3 evaluation protocol. Best results for each metric are underlined.

	Representation		Extent		Type
	Word	Character	Strict	Relaxed	Relaxed
NTER	Word2vec		78.11	89.33	85.16
NTER	Word2vec	✓	79.15	90.32	86.55
NTER	GloVe		81.35	91.90	87.65
NTER	GloVe	✓	83.09	92.28	87.92
NTER	FastText		79.09	86.82	78.76
NTER	FastText	✓	84.40	92.16	85.51
NTER	BERT		88.46	94.12	91.73
NTER	BERT	✓	<u>88.91</u>	<u>94.71</u>	<u>92.03</u>

combinations is performed. The results can be found in Table 6.3. It seems that the effect of knowledge augmentation depends both on the domain and on whether the knowledge is general, such as in KnowBERT and ManTIME resources, or purely temporal, such as in HeidelTime resources.

Table 6.3 Mean μ and standard deviation σ of change in relaxed match extent F1 score when adding different knowledge sources on TempEval-3 test data and crossvalidation on PÂTÉ with TempEval-3 evaluation protocol.

	Content		Change μ		Change σ	
	Temporal	General	TempEval-3	PÂTÉ	TempEval-3	PÂTÉ
HeidelTime Res.	✓		+0.63	+1.90	1.44	2.10
ManTIME Res.	✓	✓	-0.17	+0.57	1.64	1.07
KnowBERT		✓	-1.41	-5.87	2.16	0.63

6.3 End-to-end Temporal Tagging

As can be seen in Table 6.4, the experimental results mostly replicate the reported results of existing systems on the TempEval-3 data set as summarized in Table 5.1, with the only significant deviation being the lower performance of ClearTK. This is consistent with the reported performance when training on the whole TempEval-3 data set, as opposed to only the human-annotated subset.

For TempEval-3, SynTime and UWTime still provide state of the art performance, with the proposed NTER system surpassing previous neural systems and edging closer to the existing rule-based systems. The general and temporal knowledge-augmented NTER systems manage to improve over the best NTER system’s performance on temporal expression recognition, however these improvements do not translate to the normalization step.

For PÂTÉ, purely data-driven systems such as ClearTK and the proposed NTER system perform much better than existing rule-based systems on the recognition subtask. All systems that use the rule-based Timen normalization score exceptionally low on temporal expression normalization.

Table 6.4 Relaxed F1 scores of temporal tagging systems on TempEval-3 test data and crossvalidation on PÂTÉ with TempEval-3 evaluation protocol. Best NTER is BERT with character embeddings. General KANTER is KnowBERT with character embeddings and ManTIME resources. Temporal KANTER is BERT with character embeddings and HeidelTime resources. Extent and value F1 scores are based on relaxed match. Best results for each metric are underlined.

	Normalization	TempEval-3		PÂTÉ	
	Timen	Extent	Value	Extent	Value
HeidelTime		90.71	78.07	85.40	44.14
SynTime	✓	<u>94.96</u>	69.06	90.51	13.30
UWTime		90.64	<u>80.90</u>	80.68	<u>52.29</u>
ClearTK	✓	85.60	61.73	<u>97.13</u>	12.08
Best NTER	✓	86.40	66.40	94.71	14.89
General KANTER	✓	87.20	64.80	88.56	10.82
Temporal KANTER	✓	86.64	64.78	94.47	13.00

Chapter 7

Discussion

In the following, the results presented in the previous chapter will be interpreted with regards to possible effects and explanations. When discussing the performance of non-deterministic systems such as the NTER system, it is important to keep in mind that every system configuration was only trained once for each data set or data set split. Specifically for LSTM-based sequence labeling architectures, it has been shown that differences in random initialization can sometimes make a difference of up to 1.0 F1 score points (Reimers and Gurevych, 2017b). That means that small differences between two configurations on a single data set do not necessarily indicate a systematic difference in performance, but might just be due to random fluctuations. Reliable conclusions can only be drawn from performance differences that are consistent across multiple data sets or configurations.

7.1 Neural Temporal Expression Recognition

As in other natural language understanding tasks, BERT representations outperform all static representations on both data sets in multiple configurations for TER. The great success of contextualized word representations such as BERT representations is generally attributed to the fact that they are able to capture the meaning of a word in the context in which it was used, which is most helpful in resolving polysemy (Wiedemann et al., 2019). By contrast, static word embeddings attempt to represent all possible meanings and aspects of a word in a fixed vector. This means that a system that uses static word representations is exposed to information about a word that may not apply in the concrete context in which it was used, which introduces an additional source for errors.

GloVe vectors have already been shown to outperform Word2vec vectors on the related sequence labeling task of NER (Pennington, Socher, and C. Manning, 2014), so the increased performance on both data sets compared to other static word representations in TER comes

as no surprise. One possible explanation for these empirical results is that since GloVe takes into account global rather than local co-occurrence statistics, it is better suited to aggregate the typically sparsely distributed occurrences of temporal expressions and named entities in web data.

The positive effect of character-level information on sequence labeling tasks in general (Yang and Y. Zhang, 2018) and on TER in particular (Xu, Laparra, and Bethard, 2019) is well documented in previous literature. One intuitive explanation for its success in temporal tagging could be the handling of words that are Out Of Vocabulary (OOV), which means they have not been assigned a vector representation during pretraining. In the experiments, pretrained word representations for the most common two million English words were provided, which cover the majority of words used in a typical text. However, entity names by design often have the property of being unique, so that the entity can be distinguished from others. Similarly, numerical temporal expressions such as the date "20-01-2021" are uncountably many, especially considering that there exist multiple regional date formats. That means that if a named entity or time expression does not have a special cultural association like "Jesus" or "9/11" do, it is likely that it will not be among the two million most common words in English. The NTER system will attempt to learn semantic representations from a randomly initialized vectors for OOV words during training, but if the words are infrequent in the train data set, the learning signal is likely not strong enough to learn a meaningful semantic representation. Character-level features produced with CNNs can provide more meaningful representations for OOV words by deriving information from character sequences within the word that have been observed before, such as "2021" in the case of "20-01-2021". BERT addresses the issue of OOV words as well by using subword tokens, but the partitioning is not as fine-grained as the character-level (Devlin et al., 2019).

OOV words could also explain why the positive impact of character embeddings is much more pronounced in the voice assistant commands domain. Since PÂTÉ consists of colloquial commands, it likely contains words and phrases that are not common in the textual web data on which the representation models were pretrained. Consequently, character-level features become more important, since more OOV words require meaningful representations. As to why providing character embeddings decreased the performance of the NTER system with Glove representations on TempEval-3, one can only speculate. The huge drop in performance seems to indicate that the system learned to rely heavily on certain character-level features during training, which did not generalize to the test set. Given that the majority of annotations in the train data is machine-generated, while the test data is annotated by human experts (UzZaman, Llorens, et al., 2013), it is conceivable that the system learned during training to exploit some character-level clues that the automatic annotation tools were conditioned on.

Qualitative Analysis of Character-level Features

In order to better understand the impact of character-level features, a qualitative analysis of the expression "3:07:35" in the TempEval-3 test set is performed. In the context of the text, the expression denotes the best time in a marathon. As one might expect from its specificity, the string "3:07:35" is not among the two million most common words and is therefore an OOV word with respect to all pretrained static word representations. Since the word also never occurs in the train data set, its randomly initialized word vector is never adjusted and does not constitute a meaningful representation. BERT and KnowBERT split the expression into the subtokens "3", ":", "07", ":", and "35", for which contextualized pretrained embeddings are provided. However, previous work on numeracy in word representations has already shown that the subword tokenization approach of BERT is not suited well to the semantic representation of digit concatenations (Wallace et al., 2019). Therefore it comes as no surprise that neither static nor contextualized word representations alone enable the NTER system to recognize the temporal expression "3:07:35". In contrast, when augmented with character-level features, the NTER system is able to detect the temporal expression "3:07:35" in combination with the static Word2Vec and the contextualized KnowBERT word representations. There is no obvious explanation as to why those two representations in particular lead to success for this expression, but presumably the system learned to rely on character-level features more strongly because Word2vec and KnowBERT on their own do not contain as much useful information as other pretrained representations.

7.2 Knowledge-augmented Neural Temporal Expression Recognition

The continued volatility of GloVe representations when combined with other representations indicates that the system learns to rely heavily on other features, perhaps because they contain more useful information than GloVe vectors. Despite the significant increase in performance when combined with ManTIME knowledge on TempEval-3, the fact that the performance can decrease just as easily when combined with another knowledge sources makes the system configuration impractical for real-world applications until it can be determined that the effects are robust and not just due to different random initializations.

In contrast, the performance of the KANTER system with BERT representations remains largely unchanged when combined with external knowledge. One possible reason for this is that the system does not learn to integrate the external knowledge, since BERT representations already contain the knowledge necessary to solve the TER task. Previous work has shown

that BERT alone already incorporates a great wealth of both general and linguistic knowledge (Petroni et al., 2019).

When compared with plain BERT representations, the performance of KnowBERT representations is disappointing. One possible reason for that might be that according to the methodology of M. E. Peters et al., only the activations of the last layer are used to form the KnowBERT representations, while BERT representations are derived from a concatenation of the last four layer activations, which allows for an increased representational capability. Furthermore, since KnowBERT contains mostly general knowledge about well-known entities from Wikipedia, it is intuitive that this knowledge would be more useful in the domain of news articles about well-known entities and less so for day-to-day voice assistant commands.

Overall, it seems that the usefulness of external knowledge is dependent on the domain of the data set. General world knowledge, as encoded in KnowBERT representations and ManTIME resources, is only useful in combination with the news data set TempEval-3, and even then it decreases performance on average. Temporal knowledge as encoded in HeidelTime resources seems to increase performance on average, but not necessarily in the best systems, presumably because the underlying BERT representations already contain knowledge about temporal tokens to some degree.

Qualitative Analysis of Knowledge Features

In order to better understand the impact of temporal knowledge features, a qualitative analysis of the semantic group "THISNEXTLAST" in the HeidelTime resources is performed. As the group name suggests, the group consists of regular expressions for the words "this", "next" and "last", which are highly indicative of temporal expressions. The potential usefulness of this feature for temporal tagging is illustrated by the fact that there are 606 temporal expressions with "this", 354 with "next", and 795 with "last" in the TempEval-3 data set. By contrast, PÂTÉ has only 22 "this" expressions, four "next" expression and no "last" expressions, which reflects both the much smaller size of PÂTÉ and its skew towards future time references identified in previous literature (Zarcone, Alam, and Kolagar, 2020).

Given the large number of train examples in the TempEval-3 data set, it is unsurprising that the NTER system learns to recognize expressions such as "last year", "next year" and even the more complex "this fiscal year", regardless which word representation type and knowledge source was used. However, for the much smaller PÂTÉ data set, only FastText representations on their own encode enough information to recognize the extent of the expression "this Monday" correctly. It is only when they are augmented with temporal knowledge that KnowBERT representations are able to identify the expression. Further

research would be necessary to provide an explanation as to what properties exactly make some word representations succeed when others do not.

Despite these successes with temporal knowledge, the potential of knowledge augmentation in TER is definitely not exhausted and will require more complex augmentation mechanisms than simple feature-concatenation in order to detect the most subtle expressions. Examples 7.1 and 7.2 illustrate the need for both general world knowledge and a semantic reasoning mechanism in order to correctly recognize and normalize some temporal expressions. To start with, these expressions contain coreferences in the form of pronouns that must be resolved in order to determine the nature of the expression. In their particular respective contexts, "his" refers to former U.S. President Barack Obama, and "our" refers to mankind as a whole. Of course one could argue that semantic resolution is only necessary for normalization and not for recognition, but both "tenure" and "age" also have alternate meanings that would not imply a temporal expression, e.g. "tenure" as in "the employment status of an academic" and "age" as in "the number of years since the birth of a person". In addition to coreference resolution, world knowledge is required to conclude that Barack Obama's tenure is a uniquely identifiable time span and that the digital age of humankind is roughly equivalent to the present at the time of writing. Neither existing temporal tagging systems nor any of the proposed systems are able to recognize these expressions, which is to be expected at the current level of sophistication. However, these examples might serve as motivation for what could theoretically be achieved in further research using temporal tagging systems augmented with general knowledge.

7.3 End-to-end Temporal Tagging

It is encouraging that on the large TempEval-3 data set, the proposed NTER system exceeds the recognition performance of ClearTK, which is based on traditional machine learning techniques such as support vector machines and logistic regression. Together with the

Example 7.1 TIMEX3-annotated temporal expression from the TempEval-3 test data set that can only be recognized and normalized with external world knowledge and complex reasoning capabilities. "PRESENT_REF" indicates a reference to the present at the time of writing.

Our <TIMEX3 type="DATE" value="PRESENT_REF"> digital </TIMEX3> age is all about bits, those precise ones and zeros that are the stuff of modern computer code.

Example 7.2 TIMEX3-annotated temporal expression from the TempEval-3 test data that can only be recognized and normalized with external world knowledge and complex reasoning capabilities. "His" refers to former U.S. President Barack Obama in this context. "P5Y" indicates a duration of five years.

During his <TIMEX3 type="DURATION" value="P5Y"> tenure </TIMEX3>, he has increasingly unleashed biting comedic barbs against his critics and political adversaries.

fact that ClearTK is the best system on the much smaller PÂTÉ data set, this fits into the well-established tradeoff that traditional machine learning techniques can learn from fewer examples than neural networks, but the performance of neural networks scales reliably with the amount of available data (Kaplan et al., 2020).

The improved performance of both general and temporal knowledge-augmented NTER in the news domain, while not sufficient for neural methods to catch up to existing systems, shows that knowledge-augmented neural networks are a viable approach to the subtask of temporal expression recognition. It is intuitively plausible that this improvement in recognition does not translate to an improvement in normalization, since expressions that can only be recognized with external knowledge, such as holidays, can often not be normalized without external knowledge.

The observation that knowledge augmentation does not seem to improve best NTER system for voice assistant commands could be explained by the fact that performance is already very high at an F1 score of 94.71, which indicates that the temporal knowledge necessary to solve this task is already contained in the contextualized BERT representations. In order to further improve this score, it might be necessary to augment the system with relevant knowledge that BERT did not have access to at train time, such as specific information about the day-to-day events the voice commands are referring to.

Finally, it is not surprising that existing rule-based systems achieve significantly lower performance on both subtasks in the voice assistant domain than in the news domain, since most of them were originally designed for the shared task TempEval-3, which is based on news data. This also affects the normalization scores of the (KA)NTER system, since it relies on the rule-based Timen system for normalization. Timen has been unable to achieve an F1 score higher than 15.0 on the PÂTÉ data set, presumably because of stark differences in both content and syntactic structure of voice assistant commands compared to the news domain.

Qualitative Analysis of Domain Differences

A qualitative analysis of the expression "the 10th of next month" is performed in order to further investigate the impact of differing domains. The expression follows a schema that is common in the domain of voice assistant commands, where speakers make up sentences on the fly. This kind of expression is almost never used in deliberately composed news texts, since it combines an absolute reference to a day with a relative reference to a month. The structure of this temporal expression is so unusual for written texts that the existing rule-based systems SynTime and HeidelTime fail to identify the correct extent, since it is different from the domain for which they were designed. In contrast, the partially data-driven systems UWTime and ClearTK as well as all NTER configurations are able to learn to recognize this kind of expression during training and consequently manage to detect this expression during evaluation. While this is only a qualitative example, it underscores the decisive advantage of data-driven systems when it comes to data sets that differ from the originally intended domain of a system.

Chapter 8

Conclusion

This work has shown that an NTER system based on the LSTM-CRF architecture with pretrained contextualized word representations can achieve TER results that are competitive with existing systems in the domain of news and exceed the performance of existing rule-based systems in the domain of voice assistant commands.

The results indicate that various word representations can benefit from enrichment with external knowledge and character-level features. Experiments with general world knowledge and temporal knowledge showed that general knowledge is only in some cases useful on news data. In contrast, temporal knowledge is useful in all domains, but does not lead to performance gains in combination with BERT representations. Further research would be necessary to determine if this is due to BERT representations already incorporating the necessary temporal knowledge to some degree.

Similarly, it would be interesting to determine in additional experiments what influence the structure and retrieval mechanism of a knowledge source have on its effectiveness, which was not investigated in this work. This might also include a more complex augmentation mechanism with coreference resolution and reasoning capabilities, which a qualitative error analysis presented in this work suggests could be necessary to recognize and normalize some temporal expressions.

On the task of end-to-end temporal tagging, existing systems still outperform neural systems in both the news and voice assistant commands domain, which can be attributed to the rigidity of the employed rule-based normalization system. The increased performance of the KANTER system can only be translated to end-to-end temporal tagging improvement if there is a standalone normalization system that is able to normalize those expressions that can only be identified with the help of external knowledge. As a start, it would be sufficient to create a data-driven normalization system that can process a more diverse range of domains than the currently available rule-based systems.

The incremental performance improvements of temporal tagging systems through knowledge augmentation presented in this work might represent a first step towards the long-term prospect of machines one day being able to understand temporal expressions that have a personalized meaning for humans. As part of the larger vision of making communication between humans and computers in day-to-day interactions more intuitive and seamless, this is an important research area that is worth further scientific investigation.

References

- Akbik, Alan, Duncan Blythe, and Roland Vollgraf (Aug. 2018). “Contextual String Embeddings for Sequence Labeling”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1638–1649. URL: <https://www.aclweb.org/anthology/C18-1139>.
- Auer, Sören et al. (2007). “DBpedia: A Nucleus for a Web of Open Data”. In: *The Semantic Web*. Ed. by Karl Aberer et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 722–735.
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *CoRR* abs/1409.0473.
- Balazevic, Ivana, Carl Allen, and Timothy Hospedales (Nov. 2019). “TuckER: Tensor Factorization for Knowledge Graph Completion”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5185–5194. DOI: 10.18653/v1/D19-1522. URL: <https://www.aclweb.org/anthology/D19-1522>.
- Bethard, Steven (June 2013). “ClearTK-TimeML: A minimalist approach to TempEval 2013”. In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 10–14. URL: <https://www.aclweb.org/anthology/S13-2002>.
- Bethard, Steven and Jonathan Parker (May 2016). “A Semantically Compositional Annotation Scheme for Time Normalization”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 3779–3786. URL: <https://www.aclweb.org/anthology/L16-1599>.
- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387310738.
- Bojanowski, Piotr et al. (2017). “Enriching Word Vectors with Subword Information”. In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. DOI: 10.1162/tacl_a_00051. URL: <https://www.aclweb.org/anthology/Q17-1010>.
- Brucato, Matteo et al. (Sept. 2013). “Recognising and Interpreting Named Temporal Expressions”. In: *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*. Hissar, Bulgaria: INCOMA Ltd. Shoumen, BULGARIA, pp. 113–121. URL: <https://www.aclweb.org/anthology/R13-1015>.
- Chang, Angel X. and Christopher Manning (May 2012). “SUTime: A library for recognizing and normalizing time expressions”. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*. Istanbul, Turkey: Euro-

- pean Language Resources Association (ELRA), pp. 3735–3740. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/284_Paper.pdf.
- Cho, Kyunghyun et al. (Oct. 2014). “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches”. In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, pp. 103–111. DOI: 10.3115/v1/W14-4012. URL: <https://www.aclweb.org/anthology/W14-4012>.
- Devlin, Jacob et al. (June 2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: <https://www.aclweb.org/anthology/N19-1423>.
- Erxleben, Fredo et al. (2014). “Introducing Wikidata to the Linked Data Web”. In: *Proceedings of the 13th International Semantic Web Conference - Part I*. ISWC ’14. Berlin, Heidelberg: Springer-Verlag, pp. 50–65. ISBN: 9783319119632. DOI: 10.1007/978-3-319-11964-9_4. URL: https://doi.org/10.1007/978-3-319-11964-9_4.
- Etcheverry, Mathias and Dina Wonsever (2017). “Time Expressions Recognition with Word Vectors and Neural Networks”. In: *24th International Symposium on Temporal Representation and Reasoning (TIME 2017)*. Ed. by Sven Schewe, Thomas Schneider, and Jef Wijsen. Vol. 90. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 12:1–12:20. URL: <http://drops.dagstuhl.de/opus/volltexte/2017/7925>.
- Filannino, Michele (2012). *Temporal expression normalisation in natural language texts*. arXiv: 1206.2010 [cs.CL].
- Filannino, Michele, Gavin Brown, and Goran Nenadic (June 2013). “ManTIME: Temporal expression identification and normalization in the TempEval-3 challenge”. In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 53–57. URL: <https://www.aclweb.org/anthology/S13-2009>.
- Ganea, Octavian-Eugen and Thomas Hofmann (Sept. 2017). “Deep Joint Entity Disambiguation with Local Neural Attention”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2619–2629. DOI: 10.18653/v1/D17-1277. URL: <https://www.aclweb.org/anthology/D17-1277>.
- Gottschalk, Simon and Elena Demidova (2019). “EventKG - the Hub of Event Knowledge on the Web - and Biographical Timeline Generation”. In: vol. 10. 6. IOS Press, pp. 1039–1070.
- Harris, Zellig (1954). “Distributional structure”. In: *Word* 10.2-3, pp. 146–162. DOI: 10.1007/978-94-009-8467-7_1. URL: https://link.springer.com/chapter/10.1007/978-94-009-8467-7_1.
- Hochreiter, Sepp (Apr. 1998). “The Vanishing Gradient Problem during Learning Recurrent Neural Nets and Problem Solutions”. In: *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 6.2, pp. 107–116. ISSN: 0218-4885. DOI: 10.1142/S0218488598000094. URL: <https://doi.org/10.1142/S0218488598000094>.
- Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.

- Jurafsky, Daniel and James H. Martin (2009). *Speech and Language Processing (2nd Edition)*. USA: Prentice-Hall, Inc. ISBN: 0131873210.
- K M, Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati (June 2018). “Learning beyond Datasets: Knowledge Graph Augmented Neural Networks for Natural Language Processing”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 313–322. DOI: 10.18653/v1/N18-1029. URL: <https://www.aclweb.org/anthology/N18-1029>.
- Kaplan, Jared et al. (2020). *Scaling Laws for Neural Language Models*. arXiv: 2001.08361 [cs.LG].
- Kingma, Diederik and Jimmy Ba (Dec. 2014). “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations*.
- Klinger, Roman and Katrin Tomanek (Dec. 2007). *Classical Probabilistic Models and Conditional Random Fields*. Tech. rep. TR07-2-013. Department of Computer Science, Dortmund University of Technology.
- Kuzey, Erdal, Vinay Setty, et al. (2016). “As Time Goes By: Comprehensive Tagging of Textual Phrases with Temporal Scopes”. In: WWW ’16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, pp. 915–925. ISBN: 9781450341431. DOI: 10.1145/2872427.2883055. URL: <https://doi.org/10.1145/2872427.2883055>.
- Kuzey, Erdal, Jannik Strötgen, et al. (2016). “Temponym Tagging: Temporal Scopes for Textual Phrases”. In: *Proceedings of the 25th International Conference Companion on World Wide Web*. WWW ’16 Companion. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, pp. 841–842. ISBN: 9781450341448. DOI: 10.1145/2872518.2889289. URL: <https://doi.org/10.1145/2872518.2889289>.
- Lacroix, Timothee, Guillaume Obozinski, and Nicolas Usunier (2020). *Tensor Decompositions for temporal knowledge base completion*. arXiv: 2004.04926 [stat.ML].
- Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 282–289. ISBN: 1558607781.
- Lample, Guillaume et al. (June 2016). “Neural Architectures for Named Entity Recognition”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 260–270. DOI: 10.18653/v1/N16-1030. URL: <https://www.aclweb.org/anthology/N16-1030>.
- Lange, Lukas et al. (July 2020). “Adversarial Alignment of Multilingual Models for Extracting Temporal Expressions from Text”. In: *Proceedings of the 5th Workshop on Representation Learning for NLP*. Online: Association for Computational Linguistics, pp. 103–109. DOI: 10.18653/v1/2020.repl4nlp-1.14. URL: <https://www.aclweb.org/anthology/2020.repl4nlp-1.14>.
- Laparra, Egoitz, Dongfang Xu, and Steven Bethard (2018). “From Characters to Time Intervals: New Paradigms for Evaluation and Neural Parsing of Time Normalizations”. In: *Transactions of the Association for Computational Linguistics* 6, pp. 343–356. DOI: 10.1162/tacl_a_00025. URL: <https://www.aclweb.org/anthology/Q18-1025>.

- Laparra, Egoitz, Dongfang Xu, Ahmed Elsayed, et al. (June 2018). “SemEval 2018 Task 6: Parsing Time Normalizations”. In: *Proceedings of The 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 88–96. DOI: 10.18653/v1/S18-1011. URL: <https://www.aclweb.org/anthology/S18-1011>.
- Lee, Kenton et al. (June 2014). “Context-dependent Semantic Parsing for Time Expressions”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 1437–1447. DOI: 10.3115/v1/P14-1135. URL: <https://www.aclweb.org/anthology/P14-1135>.
- Liao, Keng-Te et al. (Dec. 2020). “Explaining Word Embeddings via Disentangled Representation”. In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 720–725. URL: <https://www.aclweb.org/anthology/2020.acl-main.72>.
- Llorens, Hector et al. (May 2012). “TIMEN: An Open Temporal Expression Normalisation Resource”. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*. Istanbul, Turkey: European Language Resources Association (ELRA), pp. 3044–3051. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/128_Paper.pdf.
- Ma, Xuezhe and Eduard Hovy (Aug. 2016). “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1064–1074. DOI: 10.18653/v1/P16-1101. URL: <https://www.aclweb.org/anthology/P16-1101>.
- Manning, Christopher D. et al. (2014). “The Stanford CoreNLP Natural Language Processing Toolkit”. In: *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55–60. URL: <http://www.aclweb.org/anthology/P/P14/P14-5010>.
- Mazur, Pawel and Robert Dale (Oct. 2010). “WikiWars: A New Corpus for Research on Temporal Expressions”. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, pp. 913–922. URL: <https://www.aclweb.org/anthology/D10-1089>.
- Mikolov, Tomas et al. (2013). *Efficient Estimation of Word Representations in Vector Space*. URL: <http://arxiv.org/abs/1301.3781>.
- Miller, George A. (Nov. 1995). “WordNet: A Lexical Database for English”. In: *Commun. ACM* 38.11, pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: <https://doi.org/10.1145/219717.219748>.
- Miller, George A. et al. (Dec. 1990). “Introduction to WordNet: An On-line Lexical Database*”. In: *International Journal of Lexicography* 3.4, pp. 235–244. ISSN: 0950-3846. DOI: 10.1093/ijl/3.4.235. eprint: <https://academic.oup.com/ijl/article-pdf/3/4/235/9820417/235.pdf>. URL: <https://doi.org/10.1093/ijl/3.4.235>.
- Napoles, Courtney, Matthew Gormley, and Benjamin Durme (June 2012). “Annotated Gigaword”. In: pp. 95–100.
- Nickel, M. et al. (2016). “A Review of Relational Machine Learning for Knowledge Graphs”. In: *Proceedings of the IEEE* 104.1, pp. 11–33. DOI: 10.1109/JPROC.2015.2483592.
- Olex, Amy et al. (June 2018). “Chrono at SemEval-2018 Task 6: A System for Normalizing Temporal Expressions”. In: *Proceedings of The 12th International Workshop on Semantic*

- Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 97–101. DOI: 10.18653/v1/S18-1012. URL: <https://www.aclweb.org/anthology/S18-1012>.
- Paszke, Adam et al. (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). “GloVe: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: <https://www.aclweb.org/anthology/D14-1162>.
- Peters, Matthew E. et al. (Nov. 2019). “Knowledge Enhanced Contextual Word Representations”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 43–54. DOI: 10.18653/v1/D19-1005. URL: <https://www.aclweb.org/anthology/D19-1005>.
- Peters, Matthew et al. (June 2018). “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: <https://www.aclweb.org/anthology/N18-1202>.
- Petroni, Fabio et al. (Nov. 2019). “Language Models as Knowledge Bases?” In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 2463–2473. DOI: 10.18653/v1/D19-1250. URL: <https://www.aclweb.org/anthology/D19-1250>.
- Pustejovsky, J. et al. (2003). “TimeML: Robust Specification of Event and Temporal Expressions in Text”. In: *New Directions in Question Answering*.
- Pustejovsky, James et al. (Jan. 2003). “The TimeBank corpus”. In: *Proceedings of Corpus Linguistics*.
- Ratinov, Lev and Dan Roth (June 2009). “Design Challenges and Misconceptions in Named Entity Recognition”. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*. Boulder, Colorado: Association for Computational Linguistics, pp. 147–155. URL: <https://www.aclweb.org/anthology/W09-1119>.
- Reimers, Nils and Iryna Gurevych (2017a). “Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks”. In: *CoRR abs/1707.06799*. eprint: 1707.06799. URL: <http://arxiv.org/abs/1707.06799>.
- (Sept. 2017b). “Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 338–348. DOI: 10.18653/v1/D17-1035. URL: <https://www.aclweb.org/anthology/D17-1035>.
- Saurí, Roser et al. (2006). *TimeML Annotation Guidelines, Version 1.2.1*.
- Seyler, Dominic et al. (July 2018). “A Study of the Importance of External Knowledge in the Named Entity Recognition Task”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne,

- Australia: Association for Computational Linguistics, pp. 241–246. DOI: 10.18653/v1/P18-2039. URL: <https://www.aclweb.org/anthology/P18-2039>.
- Steedman, Mark (1987). “Combinatory Grammars and Parasitic Gaps”. In: *Natural Language & Linguistic Theory* 5.3, pp. 403–439. ISSN: 0167806X, 15730859. URL: <http://www.jstor.org/stable/4047583>.
- Strötgen, Jannik and Michael Gertz (Sept. 2015). “A Baseline Temporal Tagger for all Languages”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 541–547. URL: <http://aclweb.org/anthology/D15-1063>.
- Strötgen, Jannik 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, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*. Ed. by Katrin Erk and Carlo Strapparava. The Association for Computer Linguistics, pp. 321–324. URL: <https://www.aclweb.org/anthology/S10-1071/>.
- Strötgen, Jannik and Michael Gertz (2011). “WikiWarsDE: A German corpus of narratives annotated with temporal expressions”. In: *In Proceedings of the conference of the German society for computational linguistics and language technology (GSCL 2011)*, pp. 129–134.
- Strötgen, Jannik, Anne-Lyse Minard, et al. (May 2018). “KRAUTS: A German Temporally Annotated News Corpus”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: <https://www.aclweb.org/anthology/L18-1085>.
- Subramanian, Sandeep et al. (2018). *Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning*. arXiv: 1804.00079 [cs.CL].
- Suchanek, Fabian, Gjergji Kasneci, and Gerhard Weikum (Jan. 2007). “YAGO: a core of semantic knowledge”. In: pp. 697–706. DOI: 10.1145/1242572.1242667.
- UzZaman, Naushad and James Allen (July 2010). “TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text”. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, pp. 276–283. URL: <https://www.aclweb.org/anthology/S10-1062>.
- UzZaman, Naushad, Hector Llorens, et al. (June 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)*. Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 1–9. URL: <https://www.aclweb.org/anthology/S13-2001>.
- Vaswani, Ashish et al. (2017). “Attention is All You Need”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS’17*. Long Beach, California, USA: Curran Associates Inc., pp. 6000–6010. ISBN: 9781510860964.
- Verhagen, Marc et al. (July 2010). “SemEval-2010 Task 13: TempEval-2”. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, pp. 57–62. URL: <https://www.aclweb.org/anthology/S10-1010>.
- Viola, P. and Michael J. Jones (2001). “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* 1, pp. I–I.
- Wallace, Eric et al. (Nov. 2019). “Do NLP Models Know Numbers? Probing Numeracy in Embeddings”. In: *Proceedings of the 2019 Conference on Empirical Methods in*

- Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5307–5315. DOI: 10.18653/v1/D19-1534. URL: <https://www.aclweb.org/anthology/D19-1534>.
- Wang, Alex et al. (Nov. 2018). “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, pp. 353–355. DOI: 10.18653/v1/W18-5446. URL: <https://www.aclweb.org/anthology/W18-5446>.
- Wiedemann, Gregor et al. (2019). “Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings”. In: *ArXiv abs/1909.10430*.
- Wolf, Thomas et al. (Oct. 2020). “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- Wu, Yonghui et al. (Sept. 2016). “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In:
- Xu, Dongfang, Egoitz Laparra, and Steven Bethard (June 2019). “Pre-trained Contextualized Character Embeddings Lead to Major Improvements in Time Normalization: a Detailed Analysis”. In: *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 68–74. DOI: 10.18653/v1/S19-1008. URL: <https://www.aclweb.org/anthology/S19-1008>.
- Yang, Jie, Shuailong Liang, and Yue Zhang (Aug. 2018). “Design Challenges and Misconceptions in Neural Sequence Labeling”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 3879–3889. URL: <https://www.aclweb.org/anthology/C18-1327>.
- Yang, Jie and Yue Zhang (July 2018). “NCRF++: An Open-source Neural Sequence Labeling Toolkit”. In: *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, pp. 74–79. DOI: 10.18653/v1/P18-4013. URL: <https://www.aclweb.org/anthology/P18-4013>.
- Zarcone, Alessandra, Touhidul Alam, and Zahra Kolagar (May 2020). “PATE: A Corpus of Temporal Expressions for the In-car Voice Assistant Domain”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 523–530. ISBN: 979-10-95546-34-4. URL: <https://www.aclweb.org/anthology/2020.lrec-1.66>.
- Zhang, Hongming et al. (July 2019). “Knowledge-aware Pronoun Coreference Resolution”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 867–876. DOI: 10.18653/v1/P19-1083. URL: <https://www.aclweb.org/anthology/P19-1083>.
- Zhao, Sendong et al. (Nov. 2019). “GRAPHENE”. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. DOI: 10.1145/3357384.3358038. URL: <http://dx.doi.org/10.1145/3357384.3358038>.
- Zhong, Xiaoshi, Aixin Sun, and Erik Cambria (July 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)*. Vancouver, Canada: Association for Computational Linguistics, pp. 420–429. DOI: 10.18653/v1/P17-1039. URL: <https://www.aclweb.org/anthology/P17-1039>.