New Domain, Major Effort? How Much Data is Necessary to Adapt a Temporal Tagger to the Voice Assistant Domain

Touhidul Alam
Liquid Studio, Accenture
Kronberg, Germany
touhidul.alam@accenture.com

Alessandra Zarcone
HumAIn Labs, Fraunhofer IIS
Erlangen, Germany
zce@iis.fraunhofer.de

Sebastian Padó
IMS, Universität Stuttgart
Stuttgart, Germany
pado@ims.uni-stuttgart.de

Abstract

Reliable tagging of Temporal Expressions (TEs, e.g., *Book a table at L'Osteria for Sunday evening*) is a central requirement for Voice Assistants (VAs). However, there is a dearth of resources and systems for the VA domain, since publicly-available temporal taggers are trained only on substantially different domains, such as news and clinical text.

Since the cost of annotating large datasets is prohibitive, we investigate the trade-off between in-domain data and performance in DA-Time, a hybrid temporal tagger for the English VA domain which combines a neural architecture for robust TE recognition, with a parser-based TE normalizer. We find that transfer learning goes a long way even with as little as 25 in-domain sentences: DA-Time performs at the state of the art on the news domain, and substantially outperforms it on the VA domain.

1 Introduction

Many Natural Language Processing (NLP) applications rely on a temporal tagger to successfully identify and normalize temporal expressions (TEs: e.g. *seven in the evening* → *T19:00*). Examples include question answering, summarization, and information extraction (Strötgen and Gertz, 2016). Temporal tagging serves to anchor events on the temporal axis and contributes to event ordering sequences (UzZaman and Allen, 2010). This is particularly useful for Voice Assistants (VAs), that is software agents such as Apple's Siri or Amazon's Alexa, which are able to interpret spoken human queries (commands) and help their users perform simple tasks, including scheduling tasks such as *setting reminders* or *creating and editing* calendar events. For example, given the query *Delete my Monday's meeting*, a VA might have to retrieve information from a calendar corresponding to the day the user is referring to as *Monday*. In order to succeed in such tasks, VAs require a reliable temporal tagger, which can identify TEs and classify them into categories (TE recognition, for example, DATE vs. TIME) and then convert them into machine-readable canonical values (TE normalization, e.g. *seven in the evening* → *T19:00*).

The major shortcoming of current temporal taggers is arguably their domain dependence, as it is well known that NLP tools degrade on out-of-domain data. The publicly available temporal taggers (Chang and Manning, 2012; Filannino et al., 2013; Strötgen and Gertz, 2013; Lee et al., 2014) have been developed and evaluated on domain-specific datasets annotated according to the TimeML standard (Pustejovsky et al., 2003a), notably the news (Pustejovsky et al., 2003b), social media (Zhong et al., 2017), narrative (Mazur and Dale, 2010), or clinical domain (Galescu and Blaylock, 2012). In contrast, to our best knowledge, there is no existing temporal tagger optimized for the VA domain, which differs considerably from other domains: it is dominated by concise stand-alone commands, typically referring to single future events (e.g., *Add yoga to my calendar tomorrow at 6*), often outside disambiguating discourse. As a result, coreference and event ordering play a smaller role than in other domains. Also, VA queries, compared to the news domain, contain more references to the time of an event (*at 6*) and to regular event repetitions (*Wake me up every day at 7*), as well as more underspecified or vague time expressions (*Remind me to call mom later this evening*) (Rong et al., 2017; Tissot et al., 2019).

A possible solution to overcome the problem of the scarcity of tagged training data for the VA domain is to adopt a transfer learning approach (Bengio, 2011). However, this leaves open the question of what the training curve looks like: how

Proceedings of the 14th International Conference on Computational Semantics, pages 144–154
June 17–18, 2021. ©2021 Association for Computational Linguistics

```
Add my appointment at Varin Salon on
<TIMEX3 tid="t1" type="DATE" value="
    2020-04-27"> April 27th </TIMEX3>
from
<TIMEX3 tid="t2" type="TIME" value="
    2020-04-27T10:30" anchorTimeID="t1">
10:30 am </TIMEX3>
to
<TIMEX3 tid="t3" type="TIME" value="
    2020-04-27T11:30" anchorTimeID="t1">
11:30 am </TIMEX3>
<TIMEX3 tid="t4" type="DURATION"  value=
    "PT1H" beginPoint="t2" endPoint="t3"
     />
to the calendar.
```

Figure 1: TimeML example from Zarcone et al. (2020).

TE	Value Pattern (*type*)	Unit
Last summer	YYYY-SS (DATE)	Season
Last year	YYYY (DATE)	Year
This month	YYYY-MM (DATE)	Month
Next week	YYYY-WXX (DATE)	Week
Sunday the 5th	YYYY-MM-DD (DATE)	Day
7 pm tonight	YYYY-MM-DDTHH (TIME)	Hour
15 minutes later	YYYY-MM-DDTHH:MM (TIME)	Minute
At 3:07:15	YYYY-MM-DDTHH:MM:SS (TIME)	Second

Table 1: Examples of temporal units, with corresponding TE examples and their value patterns.

much data is necessary until performance "flattens out"? We investigate the performance of a temporal tagger pre-trained on news and fine-tuned on the VA domain and find that a surprisingly small amount of data (less than 100 in-domain sentences) is sufficient to achieve reasonable performance on the low-resource target domain, substantially outperforming existing systems on the VA domain.

Paper structure. We first contrast annotated data in the news and VA domain (Sec. 2). After an overview of related work (Sec. 3), we introduce DA-Time, a hybrid temporal tagger for the VA domain, which uses a neural model for TE recognition and a parsing-based model for TE normalization (Sec. 4). After describing the experimental setup (Sec. 5), we present a detailed evaluation for varying amount of target domain annotations (Sec. 6).

2 Annotation and Data

2.1 The TimeML Markup Standard

TimeML is a widely-adopted framework for annotating time, events and event relations in text following the ISO 8601 standard[1] (Pustejovsky et al., 2003a). TimeML has also been used for the influential TempEval competitions (Verhagen et al., 2007, 2010; UzZaman et al., 2013) which form the basis for most work on temporal tagging. TimeML specifies four major data structures: EVENT, TIMEX3, SIGNAL, and LINK. Among these, TIMEX3 describes TEs; EVENT, SIGNAL, and LINK describe relations among TEs. For the purposes of this study, we focus on TIMEX3 and do not take relations among events into account, as motivated by the lower significance of such relations for VAs.

TEs in TIMEX3 are classified into four *types*: DATE (e.g., *May 2nd*), TIME (e.g., *tomorrow morning*), DURATION (e.g., *an hour*), SET (e.g., *every Monday*). An example is given in Figure 1. Each TE in TIMEX3 is identified by a unique ID (*tid* attribute). TEs are assigned *values* in a normalized machine-readable format following the ISO 8601 standard. Reference date information is also included on TIME type, which refers to the date to which the TE is anchored. TEs of *type* DURATION are also tagged with a *beginPoint* and *endPoint*, corresponding to the *tid* of the two TEs the DURATION *type* expression is anchored to. As Figure 1 shows, sometimes the range of a duration remains underspecified. In this case, an *empty tag* of *type* DURATION is added. Similarly, if only the duration range and either the beginning or end point are mentioned (e.g. *Book the room from 10:30 am for two hours*), then an empty TIME *type* tag is added to indicate the missing TE. If the value of a TE is derived from the value of another one, the *anchorTimeID* attribute indicates which TE the tagged TIMEX3 is anchored to.

On a more fine-grained level, TEs can be described using *temporal units* at different levels of granularity (Strötgen and Gertz, 2016), e.g. *the 2nd week of February*, *the 2nd day of February*), *next February* (month). These units are not explicitly annotated in TIMEX3, but they can be used to identify different value patterns (see Table 1).

2.2 Datasets

We now introduce the TimeML-annotated English datasets in the source (news) and target domain (VA). Descriptive statistics are reported in Table 2.

News domain The news domain is widely studied because of the vast availability of news text, and

[1] ISO 8601 is an international standard covering the exchange of date- and time-related data

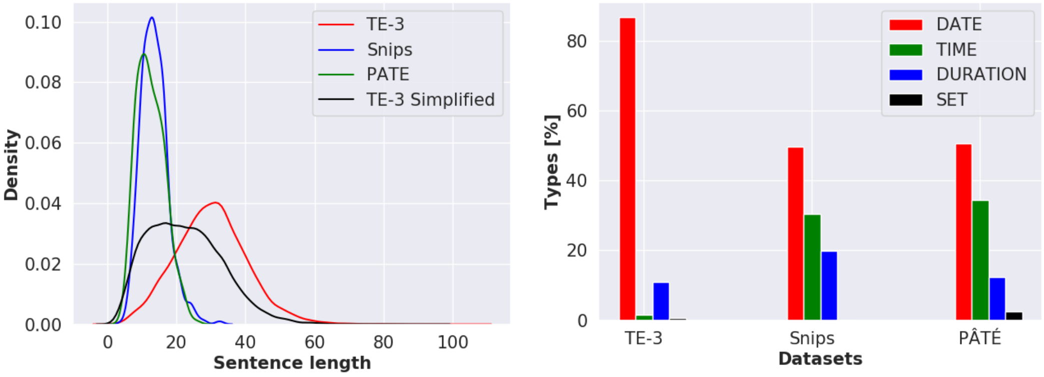

Figure 2: Comparison between the news and VA corpora on Sentence length distribution across datasets (left) and TIMEX3 type distribution (right). The Figure on the right includes empty tags for Snips and PÂTÉ.

		Tokens	Sent.s w/ TIMEXes	# of TIMEXes
News	TBAQ	99420	1469	1822
	TE-3 Silver	713091	10020	12739
	TE-3 (TBAQ+Silver)	812511	11489	14561
	TE-3 Simplified	289897	12897	14561
	TE-3 Platinum	7009	106	138
VA	Snips	9677	697	947
	PÂTÉ	5633	353	767

Table 2: Statistics on datasets for two domains (TE-3: TempEval-3[2]). TE-3 Simplified is described in 5.1.

the importance of TEs for relationships between reported events. In TempEval-3 (UzZaman et al., 2013), the manually annotated TBAQ corpus, consisting of TimeBank and AQUAINT corpus, was used as a training set (99K tokens) (Pustejovsky et al., 2003b). Additionally, a 700K-token machine-annotated corpus (TE-3 Silver) was created from Gigaword (Parker et al., 2011). Furthermore, a *platinum* set (TE-3 Platinum) was provided for evaluation, which had a higher inter-annotator agreement than existing TimeML corpora (hence the name).

Voice Assistant domain Two datasets have recently become available for the VA domain: Snips (Coucke et al., 2018) and PÂTÉ (Zarcone et al., 2020). Snips is a widely-adopted dataset for benchmarking intent and entity classification in the VA domain. No details are provided on how Snips was created. A subset of Snips was annotated with TimeML/TIMEX3 tags by Zarcone et al. (2020). PÂTÉ is a TE-rich crowdsourced dataset for the VA domain, whose collection effort was specifically focused on eliciting naturally-sounding commands containing a wide variety of TEs. As such, we focus on PÂTÉ for our final evaluation.

2.3 Cross-domain Comparison

A comparison between the news and VA domains on the basis of the abovementioned corpora is shown in Figure 2. News texts are typically grammatical and coherent reports of past events that took place at a certain moment in time. The news datasets contain longer sentences (Figure 2, left), with longer-distance relationships between events (e.g. *After that year*) that pose a challenge for normalization. VA commands, on the other hand, are comparatively shorter, and they do not provide a large sentence context nor do they typically contain references to previous event mentions. Typically, TEs in VA domain are used to refer to future events. In some cases, VA commands can contain multiple TEs, posing a challenge to the normalizer in identifying the relations among them (e.g., *Move yoga from Monday at 8 pm to Sunday at 7)*.

Figure 2 (right) shows the distribution of TIMEX3 types in the datasets. It is skewed towards DATE throughout, but DATE is even more dominant in TempEval. TIME *type* TEs are substantially underrepresented in the news domain compared to the VA domain: news are generally reported on a daily level of granularity, whereas scheduling tasks require more fine-grained temporal descriptions. Granularity differences are also reflected in the *unit* distribution: the news domain mostly contains units of type DAY (48%), while in the VA domain HOUR and DAY are equally represented as the most frequent *units* (52% DAY, 40% HOUR).

Another difference between the datasets in Figure 2 is that the VA domain datasets contain a substantial number of empty tags, which are typ-

[2] TempEval-3 Task: https://www.cs.york.ac.uk/semeval-2013/task1/index.html

ical of VA interactions where temporal information can be inferred from context (e.g., *Remind me in two hours* where the inferred absolute time information can be used to set a reminder). Snips and PÂTÉ contain around 20% and 10% empty tags respectively. In Snips, 18.6% of the DATE tags and 25.4% of the TIME (but none of the DURATION tags) are empty tags. In PÂTÉ, 91% of the DURATION tags are empty tags but only 1% of the DATE tags and 1.8% of the TIME tags are empty tags. Most of the empty tags in PÂTÉ (90%) are DURATION tags, while in Snips, they are either DATE (43%) or TIME (57%) tags. Meanwhile, the news datasets do not use empty tags in their annotation at all, so a comparison is not possible.

In sum, we can expect temporal taggers that are optimized on news to perform worse on the VA domain given the differences in distribution of types, units, and domain-specific features they rely on.

3 Related Work

The first TempEval challenge (Verhagen et al., 2007) focused on the automatic extraction of temporal relations given a TimeML-annotated dataset. TempEval-2 (Verhagen et al., 2010) introduced the task of temporal tagging of TEs for the English news domain, consisting in their recognition and normalization, and as a prerequisite for temporal information extraction, which also includes the extraction of events and of their temporal relations. TempEval-3 (UzZaman et al., 2013) extended the task to multilingual settings providing TIMEX3 annotation in English and Spanish. More recent TempEval challenges (Bethard et al., 2015, 2016, 2017) also branched out to the clinical domain. As to temporal tagging in different domains (e.g., news, narrative, colloquial, autonomic), Strötgen and Gertz (2016) addressed potential challenges, observing that existing temporal taggers work sufficiently well only in the domain they were developed for. This is probably why, to the best of our knowledge, work on temporal tagging has so far only been considered in within-domain settings.

TempEval-3 can serve as a showcase of approaches to temporal tagging. The nine participants tackled the task either with rule-based, data-driven, or hybrid methods (UzZaman et al., 2013). HeidelTime (Strötgen et al., 2013), a rule-based system, obtained the top rank. The system used regular expression-based rules to identify and normalize time expressions in multilingual settings (Strötgen and Gertz, 2015). Later, they extended their rules to cover different domains (e.g., narrative, colloquial) (Strötgen and Gertz, 2016). When TEs were underspecified (e.g. *January 6th*), domain-sensitive strategies (such as searching for contextual cues or identifying a reference time) were adopted to normalize them (e.g. to normalize *January 6th* as the previous January 6th or the forthcoming one). As rule-based systems are typically crafted to work for their reference domain, HeidelTime is not able to identify and normalize expressions that are more typical of concise commands to a VA, such as *Book a slot for the 5th*, where the month is not mentioned. UW-Time (Lee et al., 2014) is a hybrid semantic parsing-based tagger using Combinatory Categorial Grammar (Steedman and Baldridge, 2011). Compared to HeidelTime, UW-Time successfully combines hand-engineered and trained rules, showing the benefit of context-handling over rule-based approach. UW-Time can use features such as the tense of a verb to determine if the TEs refer to either the past or the future, or can determine if a four-digit number in a text refer to a year or not depending on the context. UW-Time was evaluated on the news and narrative domain and set the current state-of-the-art of temporal tagging on the TempEval-3 evaluation set, working exceptionally well but with a high degree of domain specificity.

4 DA-Time

We now present a hybrid system for temporal tagging, which we use to investigate domain adaptation of temporal tagging: DA-Time (for Domain-Adapted Time Tagger). DA-Time is a pipeline of a neural TE recognizer and a rule-based normalizer[3].

4.1 TE Recognizer

We frame TE recognition as a joint TE *type* and *unit* classification tasks. As argued in Tissot et al. (2019), temporal unit or granularity is a key feature of TEs, and can be expected to improve TE recognition, in particular for imprecise TEs[4], for example those formed by a temporal unit of a specific degree of granularity and a fuzzy quantifier (e.g., *some days, several weeks, years after*). We adopt a sequence-labelling architecture influenced by the neural NER model of Lample et al. (2016).

[3] The implementation of the TE recognizer is available at this Github repository under an academic use license: https://github.com/audiolabs/DA-Time/

[4] Since temporal unit is not an explicit part of TIMEX3, we derive it from the normalized value (details in Section 5.1).

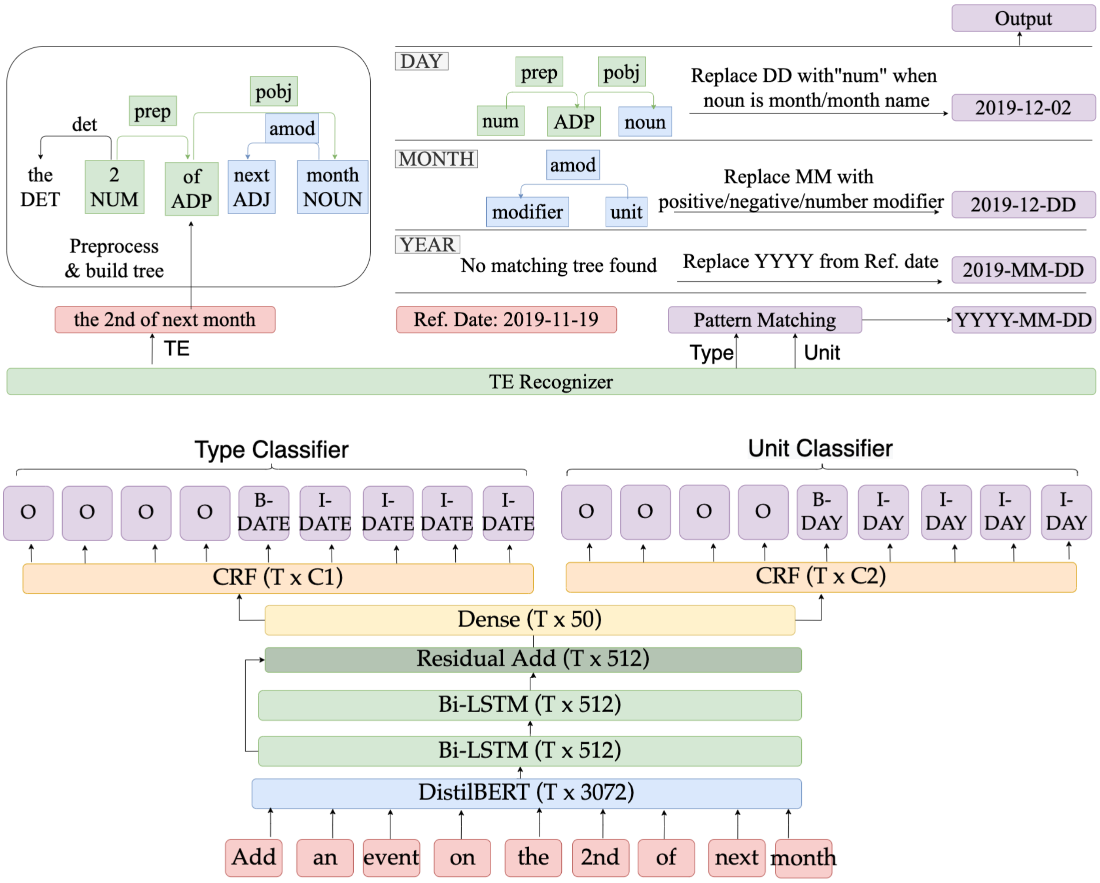

Figure 3: TE recognizer (bottom) and TE normalizer (top). Example input: *Add an event on the 2nd of next month*. Recognizer output (bottom): *the 2nd of next month* as DATE *type* and DAY *unit*. Normalizer (top): Given the recognizer output, reference date, and dependency analysis of TE, the rules are checked sequentially. The output is a normalized value for the TE.

The model takes a sentence as an input sequence and predicts *type* and *unit* in a BIO labeling scheme, as shown in Figure 3 (bottom). We use a contextualized embedding model, DistilBERT[5] (Sanh et al., 2019), as an embedding layer. DistilBERT is a smaller and faster version of BERT (Devlin et al., 2019) which is compressed during pre-training by using knowledge distillation. This improves on the inference speed compared to BERT. The embedding layer is followed by two Bi-LSTM layers. An add layer after the second Bi-LSTM which acts as a *residual add* or *skip connection* layer to improve learning (He et al., 2016). Finally, a dense layer followed by two different Conditional Random Field (CRF) layers on top is added.

Baseline model No other neural model is available as a baseline for the task of full temporal tagging of the PÂTÉ dataset, and due to its size the dataset would not be suitable for training a neural model on it. However a reasonable alternative is to adopt a pre-trained language model (Peters et al., 2018; Howard and Ruder, 2018). We propose a DistilBERT + CRF based model as a baseline, where DistilBERT is used as a pre-trained model and CRF is used to extract the labels (*type* and *unit*).

Transfer learning We apply the two approaches proposed by Felbo et al. (2017). The first method, *chain*, fine-tunes each layer sequentially (except the embedding layer in our experiment), freezing all the other layers. The second method, *full*, fine-tunes the whole network together. They found the *chain* method to perform well for sentiment analysis, as individual layers are learned with a reduced risk of overfitting. Since we observed the same pattern in preliminary experiments, we only report results from fine-tuning with the *chain* method.

For our target domain, we further apply a rule-based post-processing step to predict empty tags. Our approach consists in (1) identifying patterns of one DATE or TIME *type* begin-point (identifiable

[5] DistilBERT uncased: https://huggingface.co/distilbert-base-uncased

by tokens such as *from, between, etc.*) and one end-point (*to, and, etc.*.), where no DURATION tag is present, and (2) adding an empty DURATION tag anchored to the begin-point and end-point TEs. For example, in a command, *Set a meeting* FROM 5 TO 6 *pm*, the neural model predicts *5* and *6 pm* as two TIME *type* and further post-processing identifies an additional DURATION *type*.

4.2 TE Normalizer

For the normalization task, we propose a rule-based model using a dependency parser sketched in Figure 3 (top). TEs are fed into the parser[6]. Based on the extracted *type* and temporal *unit*, the normalizer identifies a valid normalization pattern (out of 11 expected patterns, cf. Section 5.1) for that *type* and *unit*. For example, given a DATE *type* and a WEEK *unit*, the normalizer expects to find an output pattern of YYYY-WXX. If the pattern predictions from the *type* and *unit* are incompatible (e.g., a DATE *type* with an HOUR *unit*), the normalizer uses the next most probable *unit* from the recognizer model to find as pattern that is compatible with the *unit* (e.g. a TIME *type*). This permits a more robust choice of normalization pattern and reduces the need for iterating over non-relevant rules. After identifying the pattern, each sub-unit in the pattern is normalized sequentially using parsing-based rules. In the case of YYYY-WXX, first the *value* of YEAR and then WEEK is normalized. For every pattern, we define a set of at least four rules: rules for *explicit* TEs (*12th Jan 2020*), *relative* TEs (*tomorrow morning*), *relative with modifier* (*three hours ago*), for *underspecified* TEs (*the 5th*), as well as some pattern-specific rules (e.g. for *weekly*). For each TE, the normalizer iterates over rules for each sub-unit of the pattern. Additionally, we define a gazetteer, containing the values for weekdays, times of the day, etc.

In our domain-specific settings, our normalizer assumes that underspecified expressions (e.g., *June 5*, underspecified year) refer to the past (the previous year's *June 5*) in the news domain and to the future (next year's *June 5*) in the VA domain. This hierarchically-structured rule-based model (which first identifies a pattern and then pattern-specific rules) can easily be adapted to other domains by defining different pattern-specific rules for every type of expression (relative, underspecified, etc.).

5 Experimental Setup

5.1 Data Preprocessing

We perform two data preprocessing steps: sentence simplification and inference of temporal units.

Sentence simplification As mentioned in Section 2.2, the news and VA domains greatly differ with regard to the distribution of sentence length. To reduce this discrepancy, we experiment with a parsing-based[7] text simplification method to preprocess news sentences. For each TE, it extracts the minimal complete sentence containing it (phrase type *S*). For example, in "*Washington said he will argue to save his client's life when the sentencing phase of the trial begins next Wednesday*", the underlined sub-sentence was extracted. This reduces the average length of news domain sentences from 24 to 16.

Temporal unit inference As described above, we need to access the granularity of temporal units as supervision for our model. However, temporal units are not explicitly annotated in TIMEX3: for example, *February* and *2nd week* both have type DATE but not MONTH or WEEK, respectively. However, the unit is reflected in the value pattern (XXXX-02 and XXXX-W06). Thus, we infer the TE's unit from their TimeML *value* fields using the patterns in Table 1. To cover TimeML *values* outside those mentioned in the ISO 8601, we introduce three additional *units*: QUARTER, a sub-unit of YEAR (*first quarter of 2020*); REF, which is used for reference time points (*currently*); and OTHER, which includes a number of infrequent value patterns, values for entities of type SET, and units less relevant for VAs such as century or decade.

5.2 Experiments

First, we train our DA-Time models on the news domain: DA-Time$_1$ (trained with TE-3), DA-Time$_2$ (trained with TE-3 Simplified), DA-Time$_{BL}$ (baseline model trained with TE-3). We split the dataset for our target VA domain, PÂTÉ, into a train/test set with an 80:20 ratio, keeping the class distribution constant between partitions. We perform two experiments[8]: (1) in-domain evaluation of news-trained models on the TE-3 platinum test set (all 3 DA-Time models); (2) out-of-domain evaluation

[6] We use the SpaCy dependency parser (v.2.3.0): https://spacy.io/api/dependencyparser

[7] Stanford CoreNLP parser: https://stanfordnlp.github.io/CoreNLP/

[8] (Hyper-)parameters are described in the Appendix.

Model	Training data	Extent$_{\text{strict}}$	Extent$_{\text{relax}}$	Unit$_{\text{relax}}$	Type$_{\text{relax}}$	Value$_{\text{relax}}$
HeidelTime	(rule-based)	81.8	90.7	-	83.3	78.1
UW-Time	TBAQ	83.1	91.4	-	**85.4**	**82.4**
DA-Time$_{BL}$	TE-3 (TBAQ+Silver)	81.3±1.3	87.5±1.0	74.0±0.5	74.9±2.3	59.6±1.9
DA-Time$_1$	TE-3 (TBAQ+Silver)	**86.6**±0.4	**91.4**±0.8	78.2±1.5	80.7±2.3	71.7±2.2
DA-Time$_2$	TE-3 Simplified	85.1±0.8	90.0±1.3	77.4±2.7	81.1±2.1	71.3±3.0

Table 3: Experiment 1: F1 Evaluation scores on the news domain (TempEval-3 platinum). DA-Time scores are averages of 5 runs with standard deviations.

of news-trained models on the PÂTÉ test set (DA-Time$_2$, for better comparison with the VA domain, where sentences are shorter). For our second experiment, we compare three settings: (a) direct evaluation of the news model to obtain a lower bound; (b) fine-tuning the news model on PÂTÉ-train and Snips (using Felbo et al. (2017)) and evaluating on PÂTÉ-test to obtain an upper bound; (c), repeating (b) with smaller amounts of VA data (10-100% of PÂTÉ-train with a step size of 10%, i.e., about 50 sentences) to quantify the importance of target domain data. For comparison, we report results for two existing systems, UW-Time and HeidelTime. For news, we report results from the literature, and for PÂTÉ, we evaluate the publicly available UW-Time[9] and HeidelTime[10] systems.

5.3 Evaluation Metrics

We report the F-score metrics from TempEval-3. These include (a) two measures of the overlap between the predicted and gold TE spans (*extent*), computed both in a strict (TEs are exactly matched) and a relaxed condition (TEs are partially matched); and (b) scores for attribute values (*type* and *value*) as well as *unit*. For our own system, scores are reported averages of 5 runs with standard deviations.

6 Results and Analysis

6.1 Experiment 1: In-Domain Evaluation

Table 3 shows results on the TE-3 platinum test set. For extent recognition, DA-Time$_1$ outperforms the other models, as its neural architecture benefits from the large training set. However, we also see that using the noisy silver corpus affects the *type*, and consequently the *value* scores adversely. The best-performing models for *value* scores are the rule-based HeidelTime and UW-Time, which rely on comprehensive domain-specific knowledge. The scores from the DA-Time$_{BL}$ baseline are relatively poor, which is expected here. The extension of the Bi-LSTM and residual layers in the DA-Time$_1$ allows the model to learn task-specific features. The performance of DA-Time$_2$, which uses simplified sentences, is slightly reduced - unsurprisingly, given that the test set is not simplified.

Error analysis. We observe that most errors arise from missing DURATION *type* TEs and from wrong predictions of DATE instead of DURATION. In some cases, mismatches are due to incorrect annotations in the evaluation set (e.g. a TE *2008* is annotated as DURATION but with a value of *2008*). In a few cases, DA-Time falsely predicts modifiers (e.g., *the day before*) as being part of a TE. Such modifiers are handled in the TimeML annotation by tagging them as SIGNAL - however, SIGNAL tags are out of the scope of our current work. Normalization can be further improved by leveraging on the tense of the verbs. Currently, DA-Time is built on the assumption that news texts refer to past events. In several cases the TE is underspecified, but the tense reveals it refers to a future point in time (e.g., *The event will take place on March 15*). Besides, the normalizer of DA-Time is designed to handle TEs in the VA domain. Thus, *units* like decades and centuries cannot be normalized by DA-Time.

6.2 Experiment 2: Cross-Domain Evaluation

Figure 4 shows the results for evaluating DA-Time$_2$ on the PÂTÉ test set without and with fine-tuning on various amounts of PÂTÉ and Snips data. The horizontal lines are for DA-Time$_2$ and literature models without domain adaptation.

As expected, results on PÂTÉ for models without domain adaptation are substantially worse than on the news domain. As the *Extent* and *Type* evaluations show, the strongly data-driven DA-Time$_2$

[9] UW-Time: https://bitbucket.org/kentonl/uwtime/src/master/

[10] HeidelTime (news domain): https://heideltime.ifi.uni-heidelberg.de/

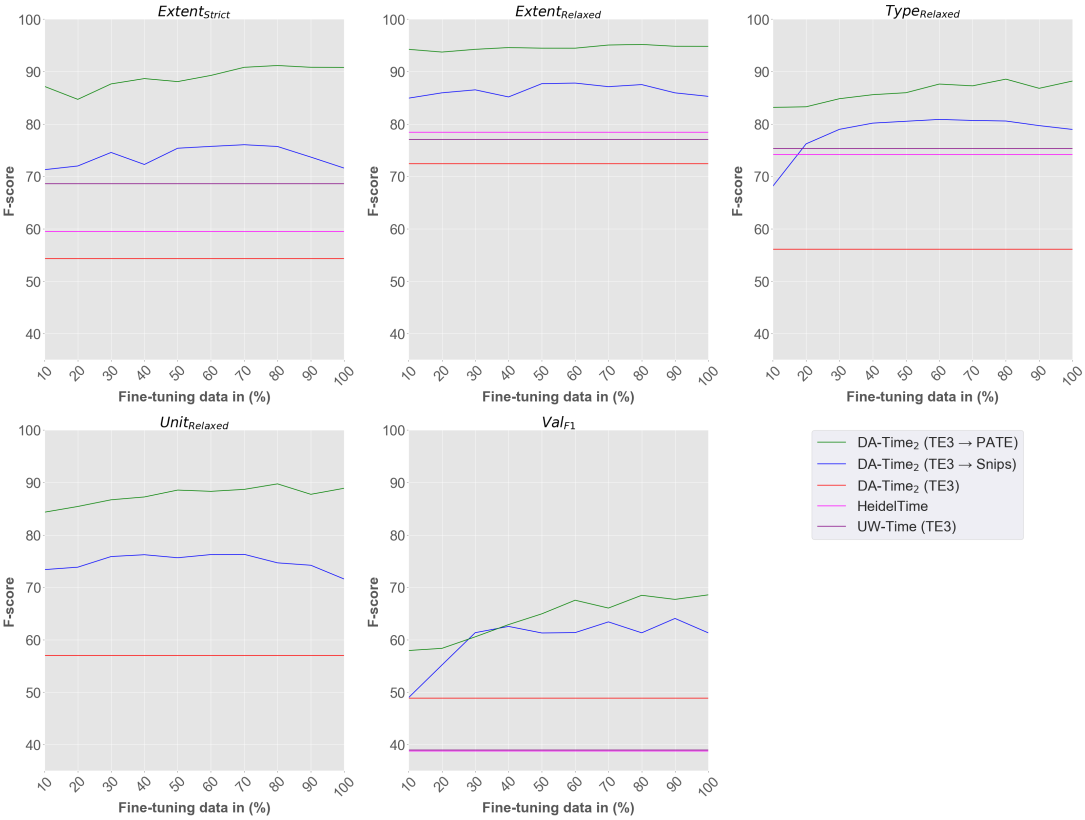

Figure 4: Experiment 2: Evaluation on PÂTÉ-test. X axis indicates the percentage of fine-tuning data used. Scores are an average of 3 runs. Horizontal lines are for models without domain adaptation. The arrows in the legend indicate which datasets were used for training and for fine-tuning, e.g. DA-Time$_2$ (TE3 → Snips) was trained with the TE3 corpus and fine-tuned with the Snips corpus. If only one dataset is indicated, the model was not fine-tuned.

TE recognizer (without fine-tuning - DA-Time$_2$ (TE3) in the figure) performs rather badly compared to HeidelTime and UWTime, presumably due to the changed properties of the input. Nevertheless, it manages to outperform both competitors in the *Value* evaluation, due to the domain-specific TE normalization component. This underlines the importance of domain specific knowledge.

Fine-tuning on Snips (DA-Time$_2$ (TE3 → Snips)) brings about notable improvement for *Extent*, *Type* and *Unit*, which also translate into an improvement for *Value*. However, the improvements flatten out after using ≈ 30% of Snips. We believe that this is due to the differences between Snips and PÂTÉ, even if the two datasets contain data from the same domain.

In comparison, fine-tuning on PÂTÉ (DA-Time$_2$ (TE3 → PÂTÉ)) yields the best results. Strikingly, the biggest jump occurs for just adding 10% of the data or about 25 sentences (strict extent: +30%, relaxed metrics (extent, *type* and *unit*): +≈ 20%, value: +10%). The figures keep improving to some extent with more data, with a final value F1 score of 68% compared to 49% without domain adaptation, and 38% for UW-Time and HeidelTime.

Error analysis. Domain adaptation improves performance in particular on minority classes. Table 4 shows a detailed class breakdown for *type* classification for one run of the model from Section 6.2. Fine-tuning with 10% of the data increases the F-score for the TIME *type* from 0 to 75%, as precision and recall increase by 70% and 79% respectively. The F-score for TIME further increases by 12 extra points after fine-tuning with the full amount of data (75% to 87%): The major difference between news and VA is the difference in class distribution which we have already seen in Figure 2. DURATION *type* expressions, which often contain empty tags and are thus dependent on TIME or DATE *type* TEs, also improve substantially.

Table 4 also shows a corresponding breakdown for *unit* classification. Among the two major *units* (DAY and HOUR), F-score of HOUR *unit* shows an increment of 71 and 80 points when fine-tuning with 10% and 100% of the data respectively. This is expected, as the class distribution difference influ-

	F-score	**Δ after fine-tuning**			**F-score**	**Δ after fine-tuning**	
	w/o fine-	**w/ 10%**	**w/ 100%**		**w/o fine-**	**w/ 10%**	**w/ 100%**
Type (freq.)	**tuning**	**data**	**data**	**Unit (freq.)**	**tuning**	**data**	**data**
DATE (68)	64.0	+20	+30	DAY (61)	66.0	+9	+26
TIME (48)	0.0	**+75**	**+87**	HOUR (44)	7.0	**+71**	**+80**
DURATION (21)	32.0	+36	**+40**	WEEK (5)	44.0	+0	-4
SET (3)	50.0	+30	+30	MONTH (3)	55.0	-5	+12

Table 4: Per-*type*$_{\text{relaxed}}$ and per-*unit*$_{\text{relaxed}}$ evaluation of DA-Time$_2$ on PÂTÉ test: F-scores without fine-tuning (TE3) and Δ after fine-tuning with 10% and 100% of the data (TE3 → PÂTÉ).

enced the *unit* distribution too. Other minor classes are again too infrequent for a reliable analysis.

The rule-based empty tag recognition in DA-Time$_2$ identifies some false positive TEs. This happens when two different TEs are present, which do not denote the beginning and end of an event but rather a change in schedule (BOOK *a schedule from 3 to 5 pm* Vs. MOVE *a schedule from 3 to 5 pm*). Domain adaptation however makes a difference compared to out-of-domain scenarios by correctly recognizing a singular numerical token as a TE (*Book a hotel reservation from May 3 to 5* or, *Set a reminder on May 3 at 5*) as they are quite common in the VA domain commands. But this is still a challenge when normalizing multiple TEs without identifying the relations among the TEs (e.g., *Change Star wars 9 from the 25th to the same time on the 24th*). We also find that our parsing-based normalizer provides a particular benefit for handling long TEs (e.g., *the 15th of next month* or *the day before last Tuesday*, etc.).

7 Conclusion

Identifying time expressions (TEs) is a crucial part of the interaction between a voice assistant (VA) and a user, but only small annotated TE corpora exist in the TE domain. In this paper, we have presented DA-Time, a hybrid model combining a neural TE recognizer with a rule-based TE normalizer, and assessed how much data is necessary to fine-tune DA-Time on the VA domain after pre-training on the much better resourced news domain.

We find that our DA-Time model, which performs competitively with the state of the art on news, can be fine-tuned very effectively on the VA domain. While, unsurprisingly, the best performance is achieved with the full target domain training set, already 10% of that dataset – some 25 sentences – is sufficient to achieve major improvements over the news-trained model. Particularly relevant is the improvement on the *Value F1* metric, i.e., the quality of the normalized TEs.

To our best knowledge, this is also the first approach to consider the granularity of temporal *unit* following the TimeML annotation and ISO 8601 standard, and to leverage it to recognize TEs in parallel with TIMEX3 *types* in a parallel setting. TIMEX3 *type* and *unit* are both crucial inputs for our hybrid normalizer. Our normalizer encodes some domain-specific assumptions (e.g., about underspecified TEs). These are particularly important in handling long TEs. While our normalizer is domain-specific, leveraging on temporal units can ease domain adaptation to new domains.

We believe that the small amount of necessary data for fine-tuning is promising for the generalization of temporal taggers for other specific domains. In the future, further improvement may be brought by leveraging anchored time information to identify relations among TEs. Taking into consideration of other TimeML tags (EVENT, SIGNAL) can improve some of the current limitations of the model (for example by identifying event-time relationships or prepositional modifiers). More generally speaking, training temporal taggers in a more end-to-end fashion is a promising direction that appears particularly feasible in the Voice Assistant domain. Considering DA-Time as a baseline model could lead to further neural-based research in the VA domain or for other application domains where identification of temporal information is important.

Acknowledgements

This research was carried out while the first author was affiliated with the Fraunhofer IIS. It was funded in part by the German Federal Ministry for Economic Affairs and Energy (BMWi) through the SPEAKER project (FKZ 01MK19011).

References

Yoshua Bengio. 2011. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*, UTLW'11, page 17–37. JMLR.org.

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

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

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

Angel X. Chang and Christopher Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.

Michele Filannino, Gavin Brown, and Goran Nenadic. 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)*, pages 53–57. Association for Computational Linguistics.

Lucian Galescu and Nate Blaylock. 2012. A corpus of clinical narratives annotated with temporal information. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 715–720. ACM.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 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*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1437–1447, Baltimore, Maryland. Association for Computational Linguistics.

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

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword Fifth Edition LDC2011T07. Web Download. *Linguistic Data Consortium*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 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*, pages 2227–2237. Association for Computational Linguistics.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. TimeML: Robust specification of event and temporal expressions in text.

In *Proceedings of IWCS-5, fifth International Workshop on Computational Semantics*.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, et al. 2003b. The TimeBank corpus. *Corpus Linguistics*, 2003:40.

Xin Rong, Adam Fourney, Robin N Brewer, Meredith Ringel Morris, and Paul N Bennett. 2017. Managing uncertainty in time expressions for virtual assistants. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 568–579. ACM.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *The 7th Workshop on Energy Efficient Machine Learning and Cognitive Computing (EMC^2)*.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and explicit models of grammar*, pages 181–224.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

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

Jannik Strötgen and Michael Gertz. 2016. *Domain-Sensitive Temporal Tagging*. Morgan & Claypool Publishers.

Jannik Strötgen, Julian Zell, and Michael Gertz. 2013. HeidelTime: Tuning English and developing Spanish resources for TempEval-3. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 15–19, Atlanta, Georgia, USA. Association for Computational Linguistics.

Hegler Tissot, Marcos Didonet Del Fabro, Leon Derczynski, and Angus Roberts. 2019. Normalisation of imprecise temporal expressions extracted from text. *Knowledge and Information Systems*, 61(3):1361–1394.

Naushad UzZaman and James F Allen. 2010. Event and temporal expression extraction from raw text: First step towards a temporally aware system. *International Journal of Semantic Computing*, 4(04):487–508.

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

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic. Association for Computational Linguistics.

Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden. Association for Computational Linguistics.

Alessandra Zarcone, Touhidul Alam, and Zahra Kolagar. 2020. PÂTÉ: A corpus of temporal expressions for the in-car voice assistant domain. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 516–523. ELRA.

Xiaoshi Zhong, Aixin Sun, and Erik Cambria. 2017. Time expression analysis and recognition using syntactic token types and general heuristic rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 420–429.

A TE Recognizer

Parameter	**Value**
DA-Time$_1$ input maximum length	50
DA-Time$_2$ input maximum length	30
Batch size	32
Training epochs	30
Fine-tuning epochs	20
Initial learning rate	0.001
Fine-tuning learning rate	0.0001
Bi-LSTM dropout rate	0.5
Bi-LSTM recurrent dropout rate	0.5
DistilBERT dimensions	3072
Recurrent unit	256
Dense layer unit	50
Dense layer activation	ReLu
Optimizer	Adam
Early stopping patience	5
Validation split	0.1

Table 5: Training hyper-parameters for TE Recognizer