Implicit Temporal Reasoning for Evidence-Based Fact-Checking

Liesbeth Allein, Marlon Saelens, Ruben Cartuyvels, Marie-Francine Moens
Department of Computer Science
KU Leuven, Belgium
{liesbeth.allein, ruben.cartuyvels, sien.moens}@kuleuven.be

Abstract

Leveraging contextual knowledge has become standard practice in automated claim verification, yet the impact of temporal reasoning has been largely overlooked. Our study demonstrates that time positively influences the claim verification process of evidence-based fact-checking. The temporal aspects and relations between claims and evidence are first established through grounding on shared timelines, which are constructed using publication dates and time expressions extracted from their text. Temporal information is then provided to RNN-based and Transformer-based classifiers before or after claim and evidence encoding. Our time-aware fact-checking models surpass base models by up to 9% Micro F1 (64.17%) and 15% Macro F1 (47.43%) on the MultiFC dataset. They also outperform prior methods that explicitly model temporal relations between evidence. Our findings show that the presence of temporal information and the manner in which timelines are constructed greatly influence how fact-checking models determine the relevance and supporting or refuting character of evidence documents.¹

1 Introduction

Automatically verifying information and flagging engineered falsities have been high on the political, media, and - subsequently - research agenda for quite some time (European Commission, 2022). However, the role of time in machine-assisted fact-checking has been inadequately investigated. Time can affect the veracity of previously uttered claims and the relevance of supporting or refuting evidence. This is evident in research, for example, where newly acquired knowledge may question, confirm, or refute established facts. This study proposes to ground claims and associated evidence in time and incorporate temporal reasoning abilities in the claim verification process of computational fact-checking models (Figure 1). Here, temporal reasoning is implicit since the models are not expected to make explicit predictions about time. They instead learn from data how to leverage temporal information.

Grounding a claim or evidence document in time is a complex task. On the one hand, it can be achieved through document-level grounding, which involves positioning the entire document on a timeline based on its publication date. On the other hand, a document may discuss several events that have occurred in the past, present, or future. To fa-

¹The code of this paper is publicly available: https://github.com/Marlon668/VerificationClaimsWithTimeAttribution.
cilitate more fine-grained grounding on the content level, time expressions in the text are used to to place the document on multiple positions on a timeline. Such expressions can be explicit (e.g., 27 June 2022), implicit (e.g., Christmas 2022), and relative (e.g., mid-September), which may require additional temporal information for grounding (Strö-"{o}gen and Gertz, 2013; Leeuwenberg and Moens, 2019). In this study, we ground claims and evidence on both the document and content level. This is accomplished by extracting and normalising their publication date and in-text time expressions, and subsequently relating them in terms of distance in time. This enables fact-checking models to reason over the temporal relations between a claim and its evidence on more than one level.

Contributions This study demonstrates that reasoning over temporal aspects and relations of claims and evidence not only improves fact-checking models’ prediction performance but also influences their estimation of the relevance and the supporting/refuting character of the evidence. The effects on performance are even reinforced when claims and evidence are grounded at both the document and content level, showing the appropriateness of multi-level temporal reasoning in automated fact-checking.

2 Related Work
Automated fact-checking is usually a two-phase process consisting of claim detection/selection and claim verification (Zeng et al., 2021; Guo et al., 2022). Time is arguably important in both phases. When detecting and ultimately selecting claims to fact-check, fact-checkers heed the current interest of the public in certain topics and election cycles, and rank the claims accordingly (Allein and Moens, 2020). Moreover, many selected claims mention dates or time periods (Hidey et al., 2020). Shaar et al. (2020) looked in the past and filtered out claims that are semantically similar to previously fact-checked claims to expedite the claim selection process.

While evidence-based claim verification has been widely studied (Zhong et al., 2020; Liu et al., 2020; Chen et al., 2021; Si et al., 2021; Jin et al., 2022; Xu et al., 2022; Hu et al., 2022), few studies explicitly focused on incorporating temporal reasoning in the verification process. Zhou et al. (2020) constructed (entity, value, time)-tuples representing supposedly temporal facts and verified their correctness using probabilistic graphical models. Allein et al. (2021) constrained the evidence ranking in fact-checking models on time using silver-standard evidence rankings respecting four assumptions on temporal relevance. Instead of verifying the temporal correctness of claim tuples or explicitly enforcing time-dependent evidence rankings, we let fact-checking models reason implicitly over temporal aspects of claims and evidence in natural language when checking the claims.

3 Task Description
Classifier $f$ takes a textual claim $c$ and an associated set of $N$ text documents $\{e_i\}$ serving as evidence of $c$, and returns a claim veracity label $y$.

$$f : c, \{e_i\}^N \rightarrow y$$  \hspace{1cm} (1)

To allow $f$ to reason over temporal aspects of $c$ and $e_i$, we extract and normalise publication dates and time expressions in $c$ and $e_i$, and assign them to time buckets. Temporal representations $c_t$ and $e_{i,t}$ are sequences of time bucket indices and are given as additional input to $f$:

$$f : c, c_t, \{e_i\}^N, \{e_{i,t}\}^N \rightarrow y$$  \hspace{1cm} (2)

4 Two-Level Grounding and Reasoning
To obtain temporal representations $c_t$ and $e_{i,t}$, we ground $c$ and $e_i$ in time by positioning them on a joint timeline using either their publication date $(c_t = e_{i,doc}^{doc}, e_{i,t} = e_{i,doc}^{doc})$ or in-text time expressions $(c_t = e_{i,con}^{con}, e_{i,t} = e_{i,con}^{con})$. A fact-checking model can then reason over their temporal aspects and relations at the document level or the content level, respectively (Figure 2).

4.1 Reasoning at Document Level
The publication date of $c$ serves as reference point for grounding $e_i$. This way, we lay bare the temporal relation between $c$ and $e_i$ at the document level. We adopt the approach of Allein et al. (2021) and compute the distance in days $\Delta pub \in \mathbb{Z}$ between the publication date of $c$ and that of $e_i$, where $\Delta pub < 0$ indicates that $e_i$ was published before $c$, $\Delta pub = 0$ indicates that $e_i$ and $c$ were published on the same day, and $\Delta pub > 0$ indicates that $e_i$ was published after $c$. The publication date of $e_i$ is then assigned to a time bucket $b_{pub} \in T^{doc}$ given $\Delta pub$.

Ultimately, the document-level temporal representation of $e_i$, $e_{i,t}^{doc}$, is a sequence of indices corresponding to $b_{pub}$ in $T^{doc}$, with $|e_{i,t}^{doc}| = 1$ since $e_i$
Claim: From Mid-September, Australians will begin receiving survey forms for the Australian Marriage Law Postal Survey, which the Government has commissioned in place of plebiscite.
Published: 30 November 2016

Report on the conduct of the Australian Marriage Law Postal Survey
Nov 7, 2017...Particular effort was made to ensure the survey was simple...Collection was open between 12 September 2017 and 7 November 2017.

Turnbull Government - Wikipedia
The Turnbull government, the federal executive government in Australia, led by Malcolm Turnbull, from 2015 to 2018. A September 2015 leadership spill was called, with Turnbull challenging Abbott, and...

(a) Document-Level Grounding

(b) Content-Level Grounding

Figure 2: Illustration of two-level grounding: (a) at the document level using publication dates (PD) and (b) at the content level using in-text time expressions (TE). All PD and TE are assigned to time buckets $b_{pub}$ and $b_{text}$, respectively. $\checkmark$ means that a publication date was found (only for claims) and $\times$ that no publication date was found.

4.2 Reasoning at Content Level

While the document-level approach grounds $c$ and $e_i$ as whole documents, the content-level approach places them on various positions on a timeline using time expressions found in their text. Each time expression in $e_i$ and $c$ is first extracted and normalised, and its distance in days $\Delta exp \in \mathbb{Z}$ to the publication date of $c$ is computed. They are then assigned to time buckets $b_{text} \in T^{con}$ given $\Delta exp$.

The choice of $T^{con}$ is discussed in Section 4.3. The content-level temporal representation of $e_i$ is $e_{i,t}^{con}$ is a sequence of indices where each index corresponds to a $b_{text} \in T^{con}$. The length of $e_{i,t}^{con}$ equals the number of time expressions found in $e_i$, and the $j$th element of $e_{i,t}^{con}$ corresponds to the index of the time bucket of the $j$th time expression in $e_i$. A time bucket index can occur multiple times in $e_{i,t}^{con}$. The same grounding procedure is applied to obtain content-level temporal representation $e_{i}^{con}$ for $c$. In contrast to $e_{i,t}^{doc}$, $e_{i}^{con}$ does not merely reflect the availability of a publication date for $c$ but grounds time expressions in the claim text with respect to the claim’s own publication date. The content-level grounding approach allows a fact-checking model to reason over the temporal aspects of the events discussed in $e_i$ and $c$, and their temporal relation to the publication date of $c$.

4.3 Creating Time Buckets

Time buckets $b_{pub} \in T^{doc}$ and $b_{text} \in T^{con}$ represent time intervals with respect to the publication date of $c$ (e.g., $b_{pub} = [1, 4]$ indicates that $e_i$ was published between 1 and 4 days after $c$ had been published). Following the cluster hypothesis of Jardine and van Rijsbergen (1971) which states that documents in a cluster contain similar information, the similar information in a bucket is the distance in time to $c$. For document-level grounding and reasoning, the construction and choice of $T^{doc}$ goes as follows: (1) $\Delta pub$ for each $e_i$ in the training set is computed; (2) all $\Delta pub$ are ordered in ascending order; (3) and, finally, all $\Delta pub$ are subdivided in 20 quantiles, containing a similar number of $e_i$ ($\mu = 8530.5, \sigma = 266.87$). Each quantile represents one bucket $b_{pub}$. Various numbers of quantiles...
were tested, and 20 returned the best performance on the validation set. Three buckets denoting a lacking publication date for \( e_i \), an available publication for \( c \), and a lacking publication date for \( c \) are added; hence, \( |T^{\text{doc}}| = 23 \). A similar procedure is applied for constructing \( T^{\text{con}} \) using \( \Delta_{\text{exp}} (|T^{\text{con}}| = 24, \mu = 13390.75, \sigma = 2050.4) \). However, no extra buckets \( b_{\text{text}} \) denoting (un)availability of date are added. An overview of all \( b_{\text{pub}} \) and \( b_{\text{text}} \) can be found in Appendix A. Note that the intervals of \( b_{\text{pub}} \) and \( b_{\text{text}} \) become smaller when its bounds approach 0, allowing for more fine-grained reasoning for evidence published around or at the same time as the claim. Time buckets approaching 0 (i.e., \( e_i \) situates around the same time as \( c \)) have smaller intervals than those far from 0, with even a dedicated time bucket for those evidence published or discussing events happening on the same day as the claim. The advantage of using such time buckets is that the model is more robust against bias towards larger buckets. In the fact-checking models, each bucket corresponds to a unique embedding stored in a randomly-initialised time embedding matrix, which is updated during model training.

5 Methodology

5.1 Fact-Checking Model

We take the Joint Veracity Prediction and Evidence Ranking model introduced in Augenstein et al. (2019) as base model (Figure 3). Taking \( c \) and \( e_i \) represented by their word embeddings \( w \in \mathbb{R}^{D_1} \), the text encoder encodes them to their latent representations \( h(c) \) and \( h(e_i) \in \mathbb{R}^{D_2} \). Metadata \( m \) linked to \( c \) is encoded in parallel, yielding \( g(m) \). Next, \( h(c), h(e_i), \) and \( g(m) \) are combined into a joint claim-evidence representation \( s_i \) using the matching approach introduced by Mou et al. (2016):

\[
s_i = [h(c); h(e_i); h(c) - h(e_i); h(c) \cdot h(e_i); g(m)]
\]

with \([;]\) denoting concatenation, and \([ ; ]\) the dot product. The evidence scorer projects each \( s_i \) to \( o_i \in \mathbb{R} \) forming evidence score vector \( o \in \mathbb{R}^N \). The label scorer projects each \( s_i \) to its label score vector \( q_j \in \mathbb{R}^L \) forming scoring matrix \( Q \in \mathbb{R}^{N \times L} \), with \( L \) the number of veracity labels. \( o^T \cdot Q \) gives a final score vector for all labels \( L \), to which a softmax is applied to obtain a probability distribution over all veracity labels.

![Figure 3: Overview of the fact-checking model, where temporal information on claim and evidence (in blue) is integrated before text encoding (local level; in green) or after text encoding (global level; in yellow).](image)

5.2 Incorporating Temporal Reasoning

Temporal representations \( c_t \) and \( e_{i,t} \) are transformed to their time embeddings, \( \hat{c}_t \) and \( \hat{e}_{i,t} \), and given as additional model input. The embedding dimensions depend on the stage at which they are integrated in the model.

**Local integration** \( c_t \) and \( e_{i,t} \) are integrated before encoding claim and evidence \( c \) and \( e_i \) to \( h(c) \) and \( h(e_i) \). Time embeddings \( \hat{c}_t, \hat{e}_{i,t} \) for each time bucket index in \( c_t \) and \( e_{i,t} \) are taken from the embedding matrix and projected onto the same dimension as the word embeddings \( w \in \mathbb{R}^{D_1} \) of tokens in \( c \) and \( e_i \) using a linear transformation layer \( l \).

For **document-level reasoning** (DL\(_{\text{doc}}\), eq. 4), the embeddings (there is max. one publication date; hence one time embedding per document) are prepended to those of \( c \) and \( e_i \). These are then sent to the text encoder.

\[
c = [l(\hat{c}_t^{\text{doc}}); w_0, \ldots, w_{|c|}]
\]

\[
e_i = [l(\hat{e}_{i,t}^{\text{doc}}); w_0, \ldots, w_{|e_i|}]
\]
For content-level reasoning (CLloc, eq. 5), local integration is more complex. Firstly, \( c_i^{con} \) and \( e_{i,t}^{con} \) may refer to more than one time bucket as there may be more than one time expression in \( c \) and \( e_i \). Secondly, the position of a time expression and the predicate it belongs to may provide rich information about a mentioned event. We first identify the type of each token in \( c \) and \( e_i \) (see Table 1).

<table>
<thead>
<tr>
<th>Position, predicate, and time expression marking</th>
<th>Time</th>
<th>Doc</th>
<th>Make</th>
<th>Storm</th>
<th>Landfall</th>
<th>B-PRED</th>
<th>B-TIME</th>
<th>Year-Month-Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>Month</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>Day</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
</tr>
</tbody>
</table>

Table 1: Additional sentence preprocessing when integrating \( c_i^{con} \) and \( e_{i,t}^{con} \) at the local level. The predicate \( \text{(PRED)} \) and the time expressions \( \text{(TIME)} \) are marked, with B indicating their first token, and the token distance between B-TIME and B-PRED is computed.

We then introduce three new embeddings: predicate embedding \( pr \in \mathbb{R}^{D_1} \) marks the predicate, position embedding \( po \in \mathbb{R}^{D_1} \) marks the position of the predicate, and expression embedding \( te \in \mathbb{R}^{D_1} \) marks the time expression. These additional embeddings are learned during training. The word embedding \( w \) of a token in \( c \) depends on that token’s type (same for \( e_i \) and \( e_{i,t}^{con} \)):

\[
w = \begin{cases} 
\gamma w + (1 - \gamma)(l(\hat{c}_i^{con}) + te) & \text{if B-TIME} \\
\gamma w + (1 - \gamma)te & \text{if TIME} \\
\gamma w + (1 - \gamma)(pr + po) & \text{if B-PRED} \\
\gamma w + (1 - \gamma)pr & \text{if PRED} \\
w & \text{otherwise}
\end{cases}
\]  

And for content-level reasoning (CLglob):

\[
\begin{align*}
    h(c) &= \alpha h(c) + (1 - \alpha)\text{Avg}(k(\hat{c}_i^{doc})) \\
    h(e_i) &= \alpha h(e_i) + (1 - \alpha)\text{Avg}(k(\hat{e}_{i,t}^{doc}))
\end{align*}
\]  

with \( \text{Avg} \) the average. We also experiment with a combination of document-level and content-level reasoning (DL+CLglob, eq. 8) where temporal information from both levels is provided to the model:

\[
\begin{align*}
    h(c) &= \alpha h(c) + \beta k(\hat{c}_i^{doc}) \\
           &+ (1 - \alpha - \beta)\text{Avg}(k(\hat{e}_{i,t}^{doc})) \\
    h(e_i) &= \alpha h(e_i) + \beta k(\hat{e}_{i,t}^{doc}) \\
           &+ (1 - \alpha - \beta)\text{Avg}(k(\hat{e}_{i,t}^{doc}))
\end{align*}
\]  

6 Experiments

6.1 Dataset
Experiments are conducted on MultiFC2 (Augenstein et al., 2019), a large-scale dataset containing 34,924 English claims from various fact-checking websites (= ‘domains’) where each claim is associated with at most 10 a posteriori retrieved Web documents (319,721 documents in total). It also provides metadata on speaker, category, tags, and linked entities regarding the claim. We refer to Augenstein et al. (2019) for a more detailed description of the data. Although other datasets for fact-checking have been proposed (Zeng et al., 2021), they either lack naturally occurring claims, publication dates, or multiple evidence documents (Thorne et al., 2018; Jiang et al., 2020; Ostrowski et al., 2021; Schuster et al., 2021). Nonetheless, the large size, wide diversity of topic and data sources, and high quality of the MultiFC dataset should be sufficient for showcasing the appropriateness of our approach.

6.2 Time Extraction and Normalisation
In this section, we discuss the procedure for extracting and normalising publication dates and in-text time expressions.

6.2.1 Publication Dates
The dataset provides the publication date of a claim as structured metadata. The date is represented as Year-Month-Day using rule-based temporal tagger HeidelTime (Ströten and Gertz, 2013). The publication date of an evidence document, however,

\( \text{The data is publicly available on CodaLab.} \)
is not given in the metadata. Since its publication date is often communicated before the ellipsis (‘...’) at the beginning of its text, we can automatically extract the date from the text (Allein et al., 2021). If we cannot extract a date at that position, we look for occurrences of ‘published’ or ‘posted’ in combination with a date. We again use HeidelTime for structuring the publication dates. In total, we obtain a publication date for 34,808 (99.67%) claims and 213,165 (66.67%) evidence documents.

6.2.2 In-Text Time Expressions

Extracting and normalising in-text time expressions is more challenging as they can be implicit or relative. Since in-text time expressions are usually not annotated in datasets used for fact-checking, we need to reside to pretrained methods for extracting them. We implement the Open Information Extraction (OIE) model of Stanovsky et al. (2018), which parses a sentence and labels its arguments. In this work, we focus on temporal arguments (ArgM – TMP). Since inaccurate use or absence of capital letters has been shown to decrease the performance of OIE models (Alam and Awan, 2018), the OIE model is expected to return a high number of inaccurate parses for capitalised news headlines – which make up a large portion of the claims in the data. We therefore implement a pretrained Named Entity Recognition (NER) model (Peters et al., 2017) to first detect people, locations, and organisations in the text. Then, the first token of each entity is capitalised while all other tokens are lowercased. Although capitalised temporal expressions such as weekdays and holidays are automatically lowercased too, we observed a higher quality of OIE parses when adopting this approach.

We normalise the extracted temporal expressions using HeidelTime. The document creation time (DCT) of a piece of information, in this study the publication date, is used as reference point for normalising in-text temporal expressions. In total, we obtain 321,278 in-text time expressions.

**Quality assessment** Implementing pretrained extraction and normalisation models inevitably introduces noise in the data. We therefore manually assess the quality of the NER, OIE, and HeidelTime models to ensure that the noise is limited. The assessment is performed on a randomly selected set of 10 claims and their accompanying evidence documents (104 in total) from the dataset, and performance is measured using precision (P), recall (R), and F1. Regarding NER, we investigate whether all entities have been recognised and completely extracted. The label correctness does not need to be evaluated. NER performance is 0.9054/0.9134/0.9094 (P/R/F1). For the OIE task, we assess whether all temporal expressions have been correctly extracted and parsed. OIE performance is 0.9608/0.5568/0.7050 (P/R/F1), indicating that while quite some time-related expressions have not been extracted, those found have been correctly parsed. Lastly, we evaluate the normalisation of the found expressions: HeidelTime performance is 0.9736/0.8409/0.9024 (P/R/F1). In all, we deem the quality of the pretrained extraction and normalisation models sufficiently high.

6.3 Experimental Setup

**Hyperparameter settings** Both $c$ and $e_i$ are tokenised⁢¹ and represented using word embeddings ($size = 300$ (BiLSTM); 768 (DistilRoBERTa)). We experiment with two neural text encoders for encoding $c$ and $e_i$: a two-layered bidirectional LSTM with skip-connections (dropout = 0.1, hidden size = 128) and a pretrained Sentence-DistilRoBERTa, which is a faster, distilled version of Sentence-RoBERTa (Sanh et al., 2019; Reimers and Gurevych, 2019). For sake of brevity, we continue to refer to this model as RoBERTa. Metadata $m$ is represented as a one-hot vector and encoded by a CNN (filter size = 3, kernel size = 3) with ReLU activation and 1D max pooling. The label scorer consists of two fully-connected layers (hidden size = 100; 50), both with ReLU activation. The evidence scorer is a fully-connected layer (hidden size = 100) with Leaky ReLU activation. All parameters except those of the pretrained RoBERTa model are initialised following a Xavier Uniform distribution. More detailed settings for reproducing the experiments, such as hyperparameter tuning, is provided in Appendix B.

**Pretraining and fine-tuning** The experiments are conducted on the disjoint, label-stratified train (80%), validation (10%), and test set (10%) provided by Augenstein et al. (2019). We adopt the pretraining and fine-tuning setup of Allein et al. (2021) to ensure transparent comparison. During pretraining, the model is trained on all 26 fact-checking domains where each domain is only presented once in each epoch (batch size = 32 (BiLSTM); sentence-transformers/all-distilroberta-v1.)
STM); 16 (RoBERTa), mitigating model bias towards larger domains. After each epoch, the batch order is randomly shuffled, and Adam with linear scheduler (LR = 1e−4 (BiLSTM)) or RMSprop (LR = 2e−4 (RoBERTa)) optimizes the model parameters using the cross-entropy loss on the prediction output. The best-performing model for each fact-checking domain is selected based on the validation loss. Each domain-specific model is then fine-tuned on only data from that domain and the best-performing model is again selected based on the validation loss.

7 Results

Table 2 reports model performance on the test set, aggregated over all domains, in terms of Micro F1 and Macro F14. The results show that the effect of temporal reasoning depends on (a) the level at which temporal information is integrated in the model (global vs. local), (b) the grounding/ reasoning level (document vs. content), and (c) the model architecture (BiLSTM vs. RoBERTa). Regarding the integration level, global integration (glob) substantially surpasses local integration (loc) for document-level reasoning (both models: .5501/.3343 → .6006/.4271 [BiLSTM]; .5640/.3357 → .6973/.5689 [RoBERTa]) and content-level reasoning (.5685/.3601 → .6882/.5744 [RoBERTa]). Regarding the temporal grounding and reasoning level, the results show that the combination setup where claim and evidence are grounded at both the document and content level (DL+CL) yields the overall highest performance for BiLSTM (.6417/.4743), while marginally improving RoBERTa by 2% Macro F1 (.5739). Lastly, temporal reasoning appears to impact the prediction performance of the less parameterised BiLSTM model more strongly than that of the Transformer-based RoBERTa model: .5520/.3239 → .6952/.5739 [BiLSTM]; .6952/.5532 → .6947/.5739 [RoBERTa]. A similar effect was observed by Allein et al. (2021), who explicitly modeled temporal relations between a claim and its evidence by constraining model parameters on evidence rankings following various assumptions on temporal relevance. This could be attributed to the expressive power of large pre-trained Transformers-based language models and the orders of magnitude of their pretraining set size.

Table 3 shows the comparison between our best performing set-up with the baseline from Augenstein et al. (2019) and the model with explicit temporal reasoning from Allein et al. (2021). Overall, our approach outperforms the baseline and the explicit temporal reasoning approach, especially on the Macro F1-score. This demonstrates the appropriateness of our implicit, two-level temporal reasoning method over an approach without temporal reasoning and one that explicitly models temporal relations using only publication dates.

8 Discussion

Weighting text and time We ran experiments with various weight values (α, β, γ) for combining the text features of a claim and its evidence with their temporal information5. Table 2 presents the best-performing weight values for each setting based on the validation loss. When reasoning over the document-level temporal relations (DL), the results suggest that higher importance should be attributed to the text of the claim and its evidence

---

4Computed using the scikit-learn Python package.

5A full overview of tested values and the tuning approach is provided in Appendix A.
Table 3: Results of our implicit temporal reasoning approach vs. the baseline results of Augenstein et al. (2019) (our implementation) and the explicit temporal reasoning method of Allein et al. (2021), with a BiLSTM and a Transformer text encoder. †: DistilBERT (Sanh et al., 2019) instead of RoBERTa.

<table>
<thead>
<tr>
<th></th>
<th>BiLSTM</th>
<th>Transformer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Micro F1</td>
<td>Macro F1</td>
</tr>
<tr>
<td>No temporal reasoning (Augenstein et al., 2019)</td>
<td>.5520</td>
<td>.3239</td>
</tr>
<tr>
<td>Explicit temporal reasoning (Allein et al., 2021)</td>
<td>.6265</td>
<td>.3673</td>
</tr>
<tr>
<td>Implicit temporal reasoning (Ours)</td>
<td>.6417</td>
<td>.4743</td>
</tr>
</tbody>
</table>

Importance of time in final prediction While we have shown that temporal reasoning strongly influences relevance estimations and label scores per evidence document, we now measure how much the time-aware fact-checking models rely on temporal information for their final veracity predictions. For this, we attribute the prediction of the models to the input using integrated gradients (Sundararajan et al., 2017). This attribution technique measures the attribution strength of text and time features on the final prediction. We focus on the base BiLSTM model and its best-performing temporal variants. Given the high dimensionality of text and time embeddings, the attribution strengths across all dimensions are summed to obtain an total attribution value for claim, evidence, and time ($c_t$ and $e_{i,t}$). Figure 4 illustrates the attribution values of a single data entry and presents the ranking of evidence text and time according to their attribution strength. The models typically attributed the prediction to both the claim and evidence, with a stronger emphasis on the collected evidence than on the claim. However, when time information was introduced, the attribution strength of claim and evidence texts strongly decreased, especially when evidence was grounded at the content level (CL/DL+CL). This indicates that time indeed influences model prediction.

Interestingly, the attribution ranking of temporal information was found to be distinct from that of the content, as demonstrated by the example in Figure 4. The publication dates that are closer to that of the claim obtain higher attribution strength than those far from the claim. In line with this, statistical correlation testing between $e_{i,t}$ and label scores $q_i$ - where each label score in $q_i$ for $e_{i,t}$ in the same bucket is compared to the label score in $q_i$ for $e_{i,t}$ in different buckets - show that evidence contained within the same time bucket tend to prefer the same prediction labels as their label rankings strongly correlate ($\rho = 0.7$). We can thus conclude that time

Impact on evidence relevance and label scores We analyse how and to which extent temporal reasoning influences a model’s assessment of the relevance ($o_i$) and supporting/refuting nature ($q_i$) of evidence in relation to a claim. Since the model computes $o_i$ and $q_i$ for each evidence document associated with the claim, a ranking of all evidence can be derived based on either $o_i$ or $q_i$.

We then measure the difference in such rankings between the base and the best-performing temporal models. Following Allein et al. (2021), we rely on the Spearman’s rank correlation $r_s$, which is a non-parametric, distribution-independent metric for computing the correlation between two rankings. The correlation between the base model and the temporal reasoning models with regard to evidence relevance ranking is very weak, with $0 < |r_s| < 0.19$ for both BiLSTM and RoBERTa. Also between the temporal models, those correlations are generally very weak. Interestingly, the impact of implicit temporal reasoning on a fact-checking model’s estimation of evidence relevance is arguably as strong as when performing explicit temporal reasoning (Allein et al., 2021). The correlations fall within the range of $0.17 < |r_s| < 0.24$. The correlations regarding label scoring ($q_i$) are comparable to those for evidence ranking, ranging from weak ($0.2 < |r_s| < 0.39$) and to very weak. We can thus conclude that a model’s estimation of the relevance and supporting/refuting nature of evidence documents is strongly influenced not only by the ability to reason over time, but also by the way a claim and its evidence are grounded on a timeline.
Claim: From November 10 2017 onwards, Australians will begin receiving survey forms for the Australian Marriage Law Postal Survey, which is the Government’s commissioned replacement of a plebiscite.

Title: Fact check: Is Australia the only advanced English speaking country without same-sex marriage?

Publication Date: November 10 2017

Label: on the thread (reasoning is 2: in the red, 4: in the black, 7: in orange, 184: in green)

(a) Ranking of evidence by attribution strength in terms of text and publication date (DL reasoning).

<table>
<thead>
<tr>
<th></th>
<th>Base</th>
<th>DL</th>
<th>CL</th>
<th>DL+CL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label distribution</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1) 5.3e-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2) 2.5e-3</td>
<td>(1) 1.1e-7</td>
<td>(1) 0.976</td>
<td>(1) 1.174</td>
<td></td>
</tr>
<tr>
<td>(3) .996</td>
<td>(2) 5.30e-3</td>
<td>(2) 1.72</td>
<td>(2) 2.266</td>
<td></td>
</tr>
<tr>
<td>Claim (text)</td>
<td>16.029</td>
<td>2.688</td>
<td>0.0613</td>
<td>0.0049</td>
</tr>
<tr>
<td>Claim (PD)</td>
<td>-</td>
<td>0.994</td>
<td>-</td>
<td>0.0296</td>
</tr>
<tr>
<td>Claim (TE)</td>
<td>-</td>
<td>0.0899</td>
<td>0.0441</td>
<td></td>
</tr>
<tr>
<td>Evidence (text)</td>
<td>5.279</td>
<td>0.4434</td>
<td>0.0007</td>
<td>0.001</td>
</tr>
<tr>
<td>Evidence (PD)</td>
<td>-</td>
<td>0.3213</td>
<td>-</td>
<td>0.008</td>
</tr>
<tr>
<td>Evidence (TE)</td>
<td>-</td>
<td>0.005</td>
<td>-</td>
<td>0.008</td>
</tr>
</tbody>
</table>

(b) Predicted label distribution and absolute attribution strengths. Note that strengths for evidence are for a single evidence document.

Figure 4: Illustration of BiLSTM (glob) attribution strengths for an example taken from MultiFC.

9 Conclusion

Grounding claims and associated evidence documents on a shared timeline and implicitly reasoning over their temporal relations noticeably improves the verification performance of automated fact-checking models. Time plays a dual role in this process, serving both as a source of information for verifying claims, as well as influencing the evaluation of the relevance and supporting or refuting nature of evidence documents. Further research may look into integrating temporal reasoning in claim detection and evidence retrieval processes or implementing even more sophisticated temporal reasoning during claim verification by examining the temporality of events discussed in a claim and their relation to the evidence.

Limitations

The limitations of this work mainly originate from the data and the use of pretrained models for grounding claims and evidence documents in time. Since the evidence documents were retrieved after the claim had been fact-checked by giving the claim verbatim to a search engine and selecting the first ten search results, their quality and relevance to the claim is not ensured. As a result, evidence-based fact-checking models risk relying on spurious signals in the evidence documents for predicting a claim’s veracity. Moreover, the evidence documents are presented as short snippets which only reflect small parts of the full Web documents. This not only affects content representation, but also limits temporal information extraction since many time expressions may have been omitted from the shortened text. Regarding temporal information extraction and normalisation, we had to rely on pretrained models to obtain temporal representations of claims and its associated evidence documents. This not only introduces noise in the input data, but also requires time-expensive preprocessing.

Ethics Statement

Automated fact-checking technology aims to assist people in distinguishing between verified and unverified content in professional contexts and during their daily information consumption. Nevertheless, the fact-checking models constructed in this paper - like all fact-checking models - should be deployed with caution and its predictions should never be taken as final without further human evaluation. Computational predictions are anything but flawless, and incorrect predictions may unjustly discredit the person or group who uttered the fact-checked statement(s).

Acknowledgements

This work was realised with the collaboration of the European Commission Joint Research Centre under the Collaborative Doctoral Partnership Agreement No 35332. The scientific output expressed does not imply a policy position of the European Commission. Neither the European Commission nor
any person acting on behalf of the Commission is responsible for the use which might be made of this publication. The research leading to this paper also received funding from the European Research Council (ERC) under Grant Agreement No. 788506. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

References


European Commission. 2022. Funded projects in the fight against disinformation.


A Time Buckets

Table 4 presents an overview of time buckets $b_{text}$ with their interval bounds used for document-level grounding, while Table 5 presents time buckets $b_{pub}$ with their interval bounds used for content-level grounding.

B Reproducibility Settings

This section contains settings for reproducing the experiments in this paper.

Computing infrastructure The BiLSTM models were trained on a Skylake processor type with one compute node, 9 cores per node, one GPU (GPU partition of Skylake) and 5 GB memory per core. The DistilRoBERTa models were trained on a CascadeLake processor type with one compute node with 4 cores per node, one GPU and 5 GB memory per core.

Average runtime Preprocessing, i.e., extraction of timex annotations via Heideltime, open information extraction (where before this a correction of uppercase characters is done via Named Entity Recognition), and construction of the dataset where claims and evidence are already put into buckets and the predicates and timexes are marked in the text of all the data took approximately 150 hours. Training a BiLSTM model for each domain took on average 45 hours, while a DistilRoBERTa model took 72 hours.
### Overview of time buckets for document-level grounding and reasoning: $b_{pub}$

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Number of evidence documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\infty$ days before the claim</td>
<td>$\infty$ days before the claim</td>
<td>8536</td>
</tr>
<tr>
<td>1596 days before the claim</td>
<td>741 days before the claim</td>
<td>8547</td>
</tr>
<tr>
<td>740 days before the claim</td>
<td>361 days before the claim</td>
<td>8528</td>
</tr>
<tr>
<td>360 days before the claim</td>
<td>146 days before the claim</td>
<td>8517</td>
</tr>
<tr>
<td>145 days before the claim</td>
<td>35 days before the claim</td>
<td>8626</td>
</tr>
<tr>
<td>34 days before the claim</td>
<td>4 days before the claim</td>
<td>8962</td>
</tr>
<tr>
<td>3 days before the claim</td>
<td>1 day before the claim</td>
<td>7549</td>
</tr>
<tr>
<td>on the same day as the claim</td>
<td>on the same day as the claim</td>
<td>8963</td>
</tr>
<tr>
<td>1 day after the claim</td>
<td>4 days after the claim</td>
<td>8735</td>
</tr>
<tr>
<td>5 days after the claim</td>
<td>24 days after the claim</td>
<td>8548</td>
</tr>
<tr>
<td>25 days after the claim</td>
<td>85 days after the claim</td>
<td>8345</td>
</tr>
<tr>
<td>86 days after the claim</td>
<td>187 days after the claim</td>
<td>8534</td>
</tr>
<tr>
<td>188 days after the claim</td>
<td>325 days after the claim</td>
<td>8551</td>
</tr>
<tr>
<td>326 days after the claim</td>
<td>498 days after the claim</td>
<td>8515</td>
</tr>
<tr>
<td>499 days after the claim</td>
<td>736 days after the claim</td>
<td>8502</td>
</tr>
<tr>
<td>737 days after the claim</td>
<td>1061 days after the claim</td>
<td>8533</td>
</tr>
<tr>
<td>1062 days after the claim</td>
<td>1436 days after the claim</td>
<td>8529</td>
</tr>
<tr>
<td>1437 dagen na de claim</td>
<td>1997 days after the claim</td>
<td>8537</td>
</tr>
<tr>
<td>1998 days after the claim</td>
<td>2605 days after the claim</td>
<td>8531</td>
</tr>
<tr>
<td>2606 days after the claim</td>
<td>$\infty$ days after the claim</td>
<td>8522</td>
</tr>
</tbody>
</table>

Table 4: Overview of time buckets $b_{pub}$ with their interval bounds.

### Overview of time buckets for content-level grounding and reasoning: $b_{text}$

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Number of evidence documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\infty$ days before the claim</td>
<td>18172 days before the claim</td>
<td>12853</td>
</tr>
<tr>
<td>18171 days before the claim</td>
<td>6295 days before the claim</td>
<td>12851</td>
</tr>
<tr>
<td>6294 days before the claim</td>
<td>2928 days before the claim</td>
<td>12856</td>
</tr>
<tr>
<td>2927 days before the claim</td>
<td>1678 days before the claim</td>
<td>12862</td>
</tr>
<tr>
<td>1677 days before the claim</td>
<td>989 days before the claim</td>
<td>12855</td>
</tr>
<tr>
<td>988 days before the claim</td>
<td>569 days before the claim</td>
<td>12863</td>
</tr>
<tr>
<td>568 days before the claim</td>
<td>323 days before the claim</td>
<td>12833</td>
</tr>
<tr>
<td>322 days before the claim</td>
<td>145 days before the claim</td>
<td>12935</td>
</tr>
<tr>
<td>144 days before the claim</td>
<td>42 days before the claim</td>
<td>12771</td>
</tr>
<tr>
<td>41 days before the claim</td>
<td>6 days before the claim</td>
<td>13191</td>
</tr>
<tr>
<td>5 days before the claim</td>
<td>1 day before the claim</td>
<td>13269</td>
</tr>
<tr>
<td>on the same day as the claim</td>
<td>on the same day as the claim</td>
<td>22966</td>
</tr>
<tr>
<td>1 day after the claim</td>
<td>8 days after the claim</td>
<td>15135</td>
</tr>
<tr>
<td>9 days after the claim</td>
<td>42 days after the claim</td>
<td>12665</td>
</tr>
<tr>
<td>43 days after the claim</td>
<td>124 days after the claim</td>
<td>12832</td>
</tr>
<tr>
<td>125 days after the claim</td>
<td>241 days after the claim</td>
<td>12739</td>
</tr>
<tr>
<td>242 days after the claim</td>
<td>378 days after the claim</td>
<td>12888</td>
</tr>
<tr>
<td>379 days after the claim</td>
<td>581 days after the claim</td>
<td>12828</td>
</tr>
<tr>
<td>582 days after the claim</td>
<td>834 days after the claim</td>
<td>12852</td>
</tr>
<tr>
<td>835 days after the claim</td>
<td>1178 days after the claim</td>
<td>12862</td>
</tr>
<tr>
<td>1179 days after the claim</td>
<td>1582 days after the claim</td>
<td>12834</td>
</tr>
<tr>
<td>1583 days after the claim</td>
<td>2134 days after the claim</td>
<td>12848</td>
</tr>
<tr>
<td>2135 days after the claim</td>
<td>2734 days after the claim</td>
<td>12848</td>
</tr>
<tr>
<td>2735 days after the claim</td>
<td>$\infty$ days after the claim</td>
<td>12842</td>
</tr>
</tbody>
</table>

Table 5: Overview of time buckets $b_{text}$ with their interval bounds.
**Number of model parameters** BiLSTM: 16,129,125 learnable parameters per model; DistilRoBERTa: 82,933,601 learnable parameters per model.

**Number of training and evaluation runs** Without parameterisation by $\alpha$, $\beta$, and $\gamma$: 150 epochs pretraining, 100 epochs fine-tuning (both BiLSTM and DistilRoBERTa). With parameterisation: 600 epochs pretraining, 300 epochs fine-tuning (BiLSTM); 800 epochs pretraining, 300 epochs fine-tuning (DistilRoBERTa).

**Hyperparameter bounds** We manually tested following combinations for $\alpha$ when integrating the time attribution vectors at the global level for document-level (DL) or content-level reasoning: $\alpha \in \{0.10, 0.25, 0.50, 0.75, 0.90\}$. Final $\alpha$-values: BiLSTM (DL\_glob): $\alpha = 0.10$; BiLSTM (CL\_glob): $\alpha = 0.25$; DistilRoBERTa (DL\_glob): $\alpha = 0.75$ (see Figure 5); DistilRoBERTa; DistilRoBERTa (CL\_glob): $\alpha = 0.10$. We tested following combinations for $\gamma$ when integrating the time attribution vectors at the local level for content-level reasoning (CL): $\gamma \in \{0.10, 0.25, 0.50, 0.75, 0.90\}$. Final $\gamma$-values: BiLSTM (CL\_loc): $\gamma = 0.50$; DistilRoBERTa (CL\_loc): $\gamma = 0.10$. We tested following combinations for $\alpha$ and $\beta$ when grounding the time attribution vectors at both the document and content level (DL+CL): [(\(\alpha = 0.20, \beta = 0.20\)), \((\alpha = 0.20, \beta = 0.35\)), \((\alpha = 0.20, \beta = 0.40\)), \((\alpha = 0.20, \beta = 0.55\)), \((\alpha = 0.20, \beta = 0.60)\), \((\alpha = 0.20, \beta = 0.75)\), \((\alpha = 0.35, \beta = 0.20)\), \((\alpha = 0.35, \beta = 0.35)\), \((\alpha = 0.35, \beta = 0.55)\), \((\alpha = 0.40, \beta = 0.20)\), \((\alpha = 0.40, \beta = 0.40)\), \((\alpha = 0.55, \beta = 0.20)\), \((\alpha = 0.55, \beta = 0.35)\), \((\alpha = 0.60, \beta = 0.20)\)]. Final $\alpha$- and $\beta$-values: BiLSTM (DL+CL\_glob): \((\alpha = 0.20, \beta = 0.40)\); DistilRoBERTa (DL+CL\_glob): \((\alpha = 0.20, \beta = 0.20)\). We performed a hyperparameter search trial of 100 epochs pretraining for each combination of hyperparameters. The criterions used to select the final hyperparameter values are the prediction performance (Micro/Macro F1) on the validation loss and the evolution of the validation loss (visualised on a plot, see Figure 5).

**Other parameters tested**

- Without linear scheduler;
- With linear scheduler with warm up;
- With linear learning scheduler;
- Learning rates: 0.001, 0.005, 0.0002 (only for RMSprop), 0.0001, 0.00001 (for pretraining and fine-tuning);
- Adam, RMSProp (Only BiLSTM), AdamW (only DistilRoBERTa);
- With weight decay: 0.001, 0.0001;
- Without weight decay.
Figure 5: Tuning $\alpha$ for DistilRoBERTa (DL\textsubscript{glob}) based on the prediction performance on the validation set (metrics: Micro/Macro F1) and the validation loss.