WIKITIDE: A WIKIPEDIA-BASED TIMESTAMPED DEFINITION Pairs Dataset

Hsuvas Borkakoty*, **Luis Espinosa-Anke***◇
*Cardiff NLP, School of Computer Science and Informatics, Cardiff University, UK
◇AMPLYFI, UK
{borkakotyh,espinosaankel}@cardiff.ac.uk

Abstract

A fundamental challenge in the current NLP context, dominated by language models, comes from the inflexibility of current architectures to "learn" new information. While model-centric solutions like continual learning or parameter-efficient fine-tuning are available, the question still remains of how to reliably identify changes in language or in the world. In this paper, we propose WikiTiDe, a dataset derived from pairs of timestamped definitions extracted from Wikipedia. We argue that such resource can be helpful for accelerating diachronic NLP, specifically, for training models able to scan knowledge resources for core updates concerning a concept, an event, or a named entity. Our proposed end-to-end method is fully automatic, and leverages a bootstrapping algorithm for gradually creating a high-quality dataset. Our results suggest that bootstrapping the seed version of WikiTiDe leads to better fine-tuned models. We also leverage fine-tuned models in a number of downstream tasks, showing promising results with respect to competitive baselines[1].

1 Introduction

Handling new information is one of the most critical (and vastly unresolved) challenges in the current NLP landscape, mostly because language models (LMs) such as BERT (Devlin et al., 2019), T5 (Raffel et al., 2020), GPT-3 (Brown et al., 2020) or PaLM (Chowdhery et al., 2022) can only learn from information they have seen during pretraining. This is an important limitation when it comes to dealing with updates in the world and language changes alike, since these updates, if not dealt with properly in an LM-centric system, can cause *temporal misalignment* (Luu et al., 2021; Lazaridou et al., 2021; Jang et al., 2022), which is especially harming in knowledge-intensive tasks, such as closed-book QA.

Unsuprisingly, thus, there is a significant body of work concerned with, for instance, updating language models by pretraining them on in-domain data (Gururangan et al., 2020), editing specific facts (De Cao et al., 2021; Zhu et al., 2020; Dai et al., 2021), continual learning (Agarwal and Nenkova, 2021; Del Tredici et al., 2018; Giulianelli et al., 2020; Dhingra et al., 2022; Loureiro et al., 2022), pre-training with an objective specifically designed to handle infusion of newly coined terms (Yu et al., 2021), or directly modifying the attention mechanism to account for temporality (Rosin and Radinsky, 2022). All these, in addition to the extensive body of work on diachronic and dynamic (contextualized and static) word embeddings (Hamilton et al., 2016a; Rudolph et al., 2016; Hamilton et al., 2016b; Rudolph and Blei, 2018; Hofmann et al., 2020).

Regardless of the method, however, a critical component of time-aware NLP is to have access to dynamically changing facts about language and the world so that LMs are exposed to them. As Jang et al. (2022) argues, collaborative resources such as Wikipedia or Wikidata can satisfy this desideratum, since they provide a dynamically updated[2] *life-long* resource. Given this, with WIKITIDE we put forward a benchmark comprised of definition pairs annotated in terms of whether they are the same or not, and if not, if this difference can be attributed to a fundamental change in that term, event or entity (as opposed to, for instance, semantic variations such as introduction of a paraphrase or stylistic nuances). We construct WIKITIDE in a weakly supervised manner via bootstrapping, and evaluate a number of LM-based baselines on the

[1] https://github.com/hsuvas/wiki_weakly_supervised_classifier-main.git

[2] According to https://en.wikipedia.org/wiki/Wikipedia:Statistics, Wikipedia is edited twice per second.

Proceedings of Recent Advances in Natural Language Processing, pages 207–216
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_023

	WikiTiDe Definitions	Label
p^{def}_{first}	"All or Nothing" is a song by German dance-pop group Milli Vanilli.	0
p^{def}_{second}	"All or Nothing" is a song by German dance-pop group Milli Vanilli.	
p^{def}_{first}	"Along the Navajo Trail" is a country/pop song, written by Dick Charles (pseudonym for Richard Charles Krieg), Larry Markes, and Edgar De Lange in 1945.	1
p^{def}_{second}	"Along the Navajo Trail" is a country/pop song, written by Dick Charles (pseudonym for Richard Charles Krieg), Larry Markes, and Eddie De-Lange in 1945.	
p^{def}_{first}	Alan Sheffield Ball (born March 29, 1985) is an American football cornerback for the Jacksonville Jaguars of the National Football League.	2
p^{def}_{second}	Alan Sheffield Ball (born March 29, 1985) is a former American football cornerback in the National Football League for the Dallas Cowboys, Houston Texans, Jacksonville Jaguars, and Chicago Bears.	

Table 1: Examples of WikiTiDe for each label. In these specific examples, there is full agreement between all ChatGPT instances that performed the annotation.

task of determining the type of difference between two timestamped definitions. Our results suggest that bootstrapping is helpful, and that this dataset can be used for both aiding in lexical semantics tasks, as well as for efficient scanning for critical updates in Wikipedia.

2 Related Work

This paper can be broadly positioned within two areas, namely lexicograhpic *definitions* (understood as a lexicographic resource but also as a high quality source of information for augmenting LMs), and *diachronic NLP*. We therefore make a clear distinction between them in the review of relevant works.

Definitions Definitions have traditionally played a crucial role in NLP and computational lexicography. As the building blocks of dictionaries and encyclopedias, they are used when the meaning of a word is sought (Navigli and Velardi, 2010), and thus the task of automatically constructing glossaries and terminologies is a well established task in NLP and Information Retrieval (Espinosa-Anke and Schockaert, 2018; Spala et al., 2019, 2020; Veyseh et al., 2020; Azarbonyad et al., 2023).

However, definitions have also been leveraged to improve the quality of NLP systems. For instance, Delli Bovi et al. (2015) and Espinosa-Anke et al. (2016) harnessed definitions to build knowledge bases by extracting semantic relations from them; Joshi et al. (2020) used definitions to provide additional context to LMs in reading comprehension tasks; Yu et al. (2021) pre-trained BERT on tasks that exploit definitions, specifically seeking to improve contextual representations of rare terms; and Xu et al. (2022) used definitions as the backbone of prompt-based taxonomy learning.

In a parallel strand of work, others have explored *definition modeling* systems (i.e., given a term and potentially some context, generate a definition) (Gadetsky et al., 2018; Zhu et al., 2019; Mickus et al., 2019, 2022; Bevilacqua et al., 2020), and these systems have been applied in tasks such as *controlled* definition modeling, e.g., jargon or varying technical complexity (August et al., 2022; Huang et al., 2022), as well as lexical semantics tasks like word sense disambiguation and word-in-context classification (Pilehvar and Camacho-Collados, 2019).

Diachronic NLP While there is agreement in that continual learning helps to mitigate the fundamental issues of temporal misalignment (Jang et al., 2022) and catastrophic forgetting (Cossu et al., 2022), the availability of benchmarks for retrieving new facts and evaluating LMs on their capacity to account for them is not overwhelming. Social media seems to be a particularly well suited domain for exploring temporal generalization, given its naturally fast-paced nature, and so we find a number of Twitter-specific benchmarks (Osborne et al., 2014; Yogatama et al., 2014). Moreover,

Algorithm 1 Collect Definition Pairs

Let P be the set of Wikipedia pages
Let D be the list of definition pairs
Let n be the desired number of definition pairs ($n = 10,000$)
Let SRP(p, tl) be a function for *selecting a random page* given a specific timeline tl
$D = \{\}$
while $|D| < n$ **do**
    Find a random $p \in P$ with timeline tl_y
    $tl_y \leftarrow$ SortYearsAscending(tl_y)
    $m \leftarrow$ FindMedian(t)
    $p_{first} \leftarrow$ SRP$(p, tl_y \leq m)$
    $p_{second} \leftarrow$ SRP$(p, tl_y \geq m)$
    $p^{def}_{first} \leftarrow$ GetDefinition(p_{first})
    $p^{def}_{second} \leftarrow$ GetDefinition(p_{second})
    $D \leftarrow D \cup \{(p^{def}_{first}, p^{def}_{second})\}$

other resources such as arXiv papers (Lazaridou et al., 2021) or Wikipedia (Jang et al., 2022) have been benchmarked for evaluating temporal generalization, as well as temporal variations of existing relation extraction datasets (Dhingra et al., 2022).

In this context, we argue that Wikipedia is indeed a valuable and underutilized resource for training and evaluating LMs on their language and knowledge update capabilities. While, as Jang et al. (2022) points out, not all changes in Wikipedia or Wikidata correspond to an actual change in the real world, we aim to alleviate this limitation by focusing on changes in definitions alone. In this way, we drastically reduce the chances of falsely confusing one superfluous change in a Wikipedia entry with a change that results in a necessary update of our understanding of a concept or entity. In what follows, we discuss how we create our seed for the WIKITIDE dataset, the algorithm for growing it, and then report on several experimental evaluation results.

3 WIKITIDE

In this section, we discuss, first, the process of retrieving candidate definition pairs for annotation. Then, we provide details about the annotation process, and finally, present examples and summary statistics, aimed to shed light on the properties of WIKITIDE.

The process of creating the required definition pairs of WIKITIDE is shown in Algorithm 1. In a nutshell, we start from the set P of Wikipedia pages, and construct, by sampling two sufficiently distant definitions (that is, the first sentence of a Wikipedia article $p \in P$), a dataset D which contains 10,000 unannotated definition pairs. After this, we randomly select 30% from D for annotation, which we perform combining the annotations of 4 instances of GPT-3 (Brown et al., 2020)[3]. The main motivation for "replacing" manual annotation with a LM is twofold. First, we posit that we can leverage the knowledge embedded in ChatGPT's parameters about well known entities, concepts and events (well known because they have a corresponding Wikipedia page). Second, recent work has shown that leveraging ChatGPT can outperform other annotation frameworks, for example Amazon Mechanical Turk (Gilardi et al., 2023). The four rounds of annotations we perform differ in the instruction, as the hyperparameters remain fixed (specifically $temperature = 0$ and $top\_p = 1$). The instruction combines a prompt and a few examples (potentially - but not always - covering all possible labels). The specific variations involve paraphrasing some of the instructions or definitions of labels, or selecting different examples[4]. As for the labels, we define our task as a 3-label classification problem, and hence the 3 different labels (and how they are described to ChatGPT) can be broadly defined as follows:

1. Class **0**: p^{def}_{first} and p^{def}_{second} essentially convey the same information, with negligible differences in terms of style.

2. Class **1**: p^{def}_{first} and p^{def}_{second} may be semantically similar but conveying analogous information, or else convey different information, however these differences cannot be attributed to a fundamental change or update in our understanding about p.

3. Class **2**: p^{def}_{first} and p^{def}_{second} are different, *and* this difference can be unequivocally attributed to some fundamental changes happening to p and/or our shared understanding of p, which changed during the period that spanned between p^{def}_{first} and p^{def}_{second}.

The final labels are selected as follows: We only select instances labeled as class **2** if all instances of

[3]Specifically, the version powering ChatGPT: `gpt-3.5-turbo`.

[4]One example of a prompt is provided in the appendix of this submission.

ChatGPT label it as such, thus ensuring the tightest possible agreement for this label, which is both the most interesting and infrequent in the dataset. Then, for the rest, we resort to the label assigned by the majority among three ChatGPT annotators, and only in case of draw, we incorporate a fourth one, which acts as referee. At the end of this process, we annotate 3,000 instances out of the 10,000 initial set, with a Fleiss-Kappa Agreement score of (Fleiss, 1971) of 24.84, which according to the literature, falls within the *fair* agreement range. Table 1 shows illustrative examples of definition pairs in WIKITIDE. This 3k *training set* (TS) has the following label distribution: 1,082 examples for label 0; 1,830 for label 1; and 87 definition pairs for the most interesting label 2. In the following section we describe how we use TS to fully annotate D.

4 Bootstrapping WIKITIDE

With TS being the ChatGPT-annotated seed dataset in WIKITIDE (with a label set $L = \{0, 1, 2\}$), let DS be the remaining unannotated 7,000 instances, and $D = TS \cup DS$. We seek to iteratively bootstrap a development set with "high confident" predictions, starting from a seed classifier trained, in a first iteration, only on TS. We argue that this approach, which can be traced back to applications in word sense disambiguation and definition extraction (Yarowsky, 1995; Espinosa-Anke et al., 2015), can be effectively applied to our use case as each newly bootstrapped definition pair will be reliable indicatives of the source training set, which can contribute to increase recall as the model will have seen more positive examples.

As summarized in Algorithm 2, the bootstrapping process requires at a minimum an annotated training set TS and an unannotated test set DS, and optionally held-out test set HS to monitor performance. At the first iteration, we set $|TS| = 2160$; $|DS| = 7,000$; and $|HS| = 840$. We then fire the bootstrapping process, in which, first, a model is trained and applied on DS, then we extract the K most confident predictions for each label, append them to TS, and remove them from DS. Every time we exhaust all labels in L, we evaluate a new instance of the model on HS.

In terms of classifier, we select a wide range of models to evaluate, all of them based on the Transformers architecture (Vaswani et al., 2017), namely BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DistilBERT and DistilroBERTa (Sanh et al., 2020), Tiny-BERT (Bhargava et al., 2021; Turc et al., 2019) and XLM-Roberta-base (Conneau et al., 2019)[5]. Finally, in terms of manipulating the inputs to these models, we opt for minimal preprocessing, simply injecting special tokens '<y>' and '</y>' for isolating timespans, and '<t>' and '</t>' in order to mark the target term.

Algorithm 2 Bootstrapping on WIKITIDE

Require: Initial training set TS
Require: Development set DS
Require: Held-out test set HS
Require: Label set $L = \{0, 1, 2\}$
Require: K $\leftarrow 10$
Require: Temperature $T > 0$
while $|DS| \geq$ topnPreds $\cdot |L|$ **do**
  model $\leftarrow$ trainModel(TS)
  model(DS) // Apply model to DS
  for $l \in L$ **do**
    $DS_l \leftarrow \{x \mid x \in DS, \text{label}(x) = l\}$
    $P_l \leftarrow \{P(x, l) \mid x \in DS_l\}$
    Sort P'_l in descending order
    $DS'_l \leftarrow$ Top K instances from DS_l based on P'_l
    $TS \leftarrow TS \cup DS'_l$
    $DS \leftarrow DS \setminus DS_l$
  evaluateModel(model, HS)

4.1 Results and Discussion

We flesh out the results obtained by different models in the task of predicting, given a pair of definitions from Wikipedia, the labels introduced in Section 3. As can be seen in Table 2, the bootstrapped models are consistently better than their base counterparts (which, we recall, are equivalent models but being trained only on TS). RoBERTa-based models are superior to the rest, and crucically, they also reach to the best performing iteration at later stages, which suggests they tend to overfit less to the training set. In terms of gap between base and boostrapped models, this is rather large, and largest for label **2**. As an example, RoBERTa-large is almost 40 points more precise when bootstrapped, and 27 F1 points better. Interestingly, our intuition of using a multilingual model to handle "foreign" (non English) spellings, typically used in Wikipedia definitions for non English entities or concepts, seems to not work well, with XLM-

[5]All of them available at the Huggingface model hub www.huggingface.co.

		Label 2			Label 1			Label 0			Avg.			
Model	**Boot.**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**BI**
roberta-base	no	48.98	50.00	49.49	77.47	77.67	77.56	78.94	79.69	79.24	68.47	69.12	68.76	
roberta-base	yes	81.22	70.35	74.58	86.93	**88.33**	87.08	88.07	**90.13**	88.43	85.41	**82.94**	83.62	47
distilbert-base-cased	no	64.83	72.44	67.78	75.96	76.98	75.81	77.79	79.33	78.05	72.88	76.25	73.88	
distilbert-base-cased	yes	74.28	64.40	68.00	80.16	81.34	80.12	81.44	83.22	81.54	78.62	76.32	76.56	28
xlm-roberta-base	no	48.99	50.00	49.49	29.88	50.00	37.41	30.89	50.00	38.19	36.59	50.00	41.70	
xlm-roberta-base	yes	67.91	58.52	61.43	84.72	86.09	84.61	86.53	88.65	86.56	79.72	77.75	77.53	9
bert-base-cased	no	60.97	60.97	60.97	59.84	53.57	48.33	65.68	54.67	49.62	62.17	56.41	52.98	
bert-base-cased	yes	63.73	**72.31**	66.89	72.24	73.12	72.07	73.60	74.83	73.74	69.86	73.42	70.90	14
bert-tiny	no	48.76	40.77	44.41	51.09	50.86	49.73	41.36	47.46	39.91	47.07	46.36	44.68	
bert-tiny	yes	50.80	52.54	50.52	57.42	57.15	57.19	57.66	56.68	56.72	55.29	55.49	54.81	44
distilroberta-base	no	48.99	50.00	49.49	73.38	73.88	71.64	75.23	76.24	73.14	65.87	66.71	64.76	
distilroberta-base	yes	60.86	66.43	63.01	80.67	81.88	80.52	83.05	84.84	83.33	74.86	77.72	75.61	11
roberta-large	no	48.99	50.00	49.49	81.03	64.34	62.86	82.19	65.15	64.57	70.74	57.17	58.97	
roberta-large	yes	**88.29**	70.47	**76.56**	**87.59**	88.25	**87.86**	**88.76**	89.90	**89.21**	**88.21**	82.87	**84.54**	54

Table 2: Results on the held-out test set HS for a number of LMs. For the bootstrapped models, we also report the best iteration (column **BI**).

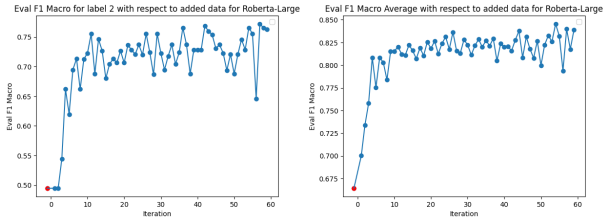

Figure 1: Macro-F1 scores of Roberta-Large with respect to Number of Iterations

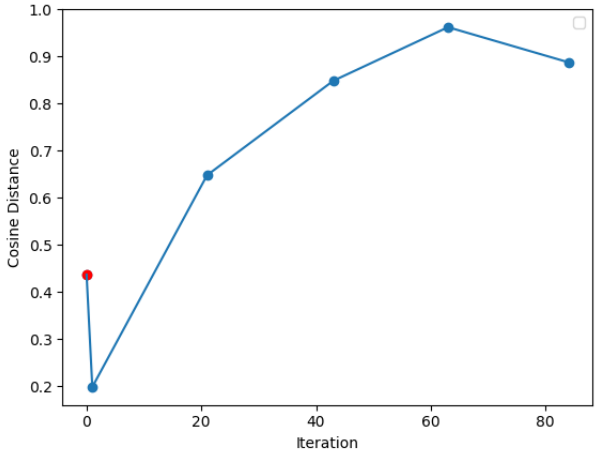

Figure 2: Cosine Distance of definition pairs for Label 2 with respect to bootstrapping iterations

roBERTa-base being the 2nd to last model, only surpassing BERT-tiny.

In terms of analyzing the bootstrapping iterative process, we can see in Figure 1 that the improvements of the bootstrapped models becomes apparent after few iterations, both for the most relevant label 2 (left plot) and on average (right plot). We also see less "up and down spikes" for the average results, suggesting that performance on the other labels becomes smoother over time. Moreover, in order to gain further understanding on the effects of the bootrsapping process into the differences in definition pairs over time, we measure *semantic drift*, i.e., whether (or, more precisely, the extent to which) the bootstrapped training set exhibits an increasingly diverse set of definitions, measured by how dissimilar they are as they are iteratively fetched from DS. We focus on label 2, and plot the results of this analysis in Figure 2, which clearly shows an increasing drift in average distances. This confirms that the bootstrapped training set is semantically more diverse than the seed ChatGPT-annotated version.

As a form of qualitative evaluation, we list in Table 3 a set of bootstrapped instances from one of the best performing models (RoBERTa-base). Note that these are not carefully selected examples, as we have simply listed an instance of high confidence classifications per label. We can see the improvement in quality of 2-labeled instances, especially between iterations 1 and 83, in which the difference in knowledge concerning Carlos Alberto Valencia is minimal in terms of string edit distance, however the model correctly identified a critical change for this named entity, specifically, the fact that he changed teams.

Iteration		WikiTiDe Definitions	Label
1	p^{def}_{first}	Argentine football saw Lomas Athletic Club win their 5th Argentine championship in 6 seasons	2
	p^{def}_{second}	Argentine football saw Lomas win its 5th Primera División championship within 6 seasons.	
1	p^{def}_{first}	The 7th Army Aviation Regiment is an army aviation formation of the Ukrainian Ground Forces	1
	p^{def}_{second}	The Army Aviation Brigade is an army aviation formation of the Ukrainian Ground Forces.	
1	p^{def}_{first}	The 16S rRNA is a long component of the small prokaryotic ribosomal subunit (30S) and is known to interact with the 50S subunit in both P and A site.	0
	p^{def}_{second}	16S ribosomal RNA (or 16S rRNA) is the RNA component of the 30S subunit of a prokaryotic ribosome (SSU rRNA).	
43	p^{def}_{first}	Dr. Bhupendranath Dutta was a famous Indian revolutionary and later a noted Sociologist.	2
	p^{def}_{second}	Bhupendranath Datta was an Indian revolutionary and later a noted sociologist and anthropologist.	
43	p^{def}_{first}	Dexia Mons-Hainaut is the Belgian professional basketball club, who based in Quaregnon.	1
	p^{def}_{second}	Belfius Mons-Hainaut is a Belgian professional basketball club that is based in Mons, Wallonia.	
43	p^{def}_{first}	Berkshire soil series is the name given to a well drained loam or sandy loam soil which has developed on glacial till in parts of southern Quebec, eastern New York State and New England south to Massachusetts.	0
	p^{def}_{second}	Berkshire soil series is the name given to a well-drained loam or sandy loam soil which has developed on glacial till in parts of southern Quebec, eastern New York State and New England south to Massachusetts.	
83	p^{def}_{first}	Carlos Alberto Valencia is a Colombian left wing back who plays for River Plate of Buenos Aires, Argentina.	2
	p^{def}_{second}	Carlos Alberto Valencia Paredes is a Colombian footballer who plays as a left-back for Independiente Medellín.	
83	p^{def}_{first}	The Carnegie Free Library of Beaver Falls was the first public library built in Beaver County, Pennsylvania.	1
	p^{def}_{second}	The Carnegie Free Library of Beaver Falls is a historic Carnegie library in the city of Beaver Falls, Pennsylvania, United States.	
83	p^{def}_{first}	Carl-Johan Lindqvist is a Swedish luger who competed in the early 1990s	0
	p^{def}_{second}	Carl-Johan Alexander Lindqvist (born November 15, in Tyresö) is a Swedish luger who competed in the early 1990s.	

Table 3: Examples of Model output on different iterations of Bootstrapping for Roberta-Base.

5 Case Study: WiC-TSV

The WiC-TSV (Word in Context-Target Sense Verification) task (Breit et al., 2021) is a "shootoff" from the original WiC task (Pilehvar and Camacho-Collados, 2019). It proposes a binary classification problem, where the input is a pair of sentences: the first one, a sentence with a target word in context, and the second one, a definition of that target word. This is a suitable test bet for a model fine-tuned on WIKITIDE, since this is a dataset which essentially measures definition similarity. However, since WIKITIDE is a multilabel dataset, we combine labels 1 and 2 as label 0 in WiC-TSV and assume equivalence between the notion of "change" in WIKITIDE and polysemy in WiC-TSV. For our model to work, both input sentences must be definitions, however, this is not always the case in WiC-TSV. To work around this limitation, we replace the non-definition sentences in WiC-TSV with a definition generated using ChatGPT (Brown et al., 2020). Both sets of results (directly applying our model to WiC-TSV as well as replacing one of its sentences with a ChatGPT-generated definition) are reported, for train, test and development sets[6] (which is possible as we cast this problem as an unsupervised classification task), in Table 4.

[6] https://github.com/semantic-web-company/wic-tsv/tree/master/data/en.

	Train				Dev				Test			
	Original		GPT3.5		Original		GPT3.5		Original		GPT3.5	
	Base	Bootsr.	Base	Bootsr.	Base	Bootsr.	Base	Bootsr.	Base	Bootsr.	Base	Bootsr.
roberta-base	0.33	0.48	0.33	0.35	0.34	0.44	0.33	0.34	0.34	0.50	0.34	0.35
distilbert-base-cased	0.33	0.46	0.33	0.33	0.3	0.47	0.34	0.34	0.34	0.43	0.34	0.34
xlm-roberta-base	0.33	0.39	0.33	0.40	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.43
bert-base-cased	0.3	0.38	0.33	0.39	0.46	0.52	0.3	0.34	0.48	0.48	0.34	0.43
bert-tiny	0.33	0.33	0.33	0.35	0.3	0.34	0.33	0.34	0.36	0.36	0.37	0.35
distilroberta-base	0.34	0.34	0.33	0.34	0.34	0.34	0.33	0.34	0.36	0.36	0.34	0.35
roberta-large	0.30	0.53	0.33	0.50	0.34	0.51	0.34	0.48	0.34	0.45	0.34	0.45

	Train				Dev				Test			
	Original		GPT3.5		Original		GPT3.5		Original		GPT3.5	
	Base	Bootsr.	Base	Bootsr.	Base	Bootsr.	Base	Bootsr.	Base	Bootsr.	Base	Bootsr.
roberta-base	0.50	0.50	0.50	0.50	0.51	0.48	0.50	0.51	0.51	0.53	0.51	0.51
distilbert-base-cased	0.5	0.49	0.49	0.50	0.51	0.50	0.51	0.50	0.51	0.48	0.50	0.51
xlm-roberta-base	0.50	0.50	0.50	0.49	0.5	0.51	0.50	0.51	0.50	0.51	0.51	0.50
bert-base-cased	0.50	0.52	0.50	0.52	0.50	0.50	0.51	0.51	0.49	0.49	0.51	0.51
bert-tiny	0.50	0.50	0.50	0.50	0.51	0.51	0.40	0.5	0.49	0.50	0.48	0.50
distilroberta-base	0.50	0.50	0.50	0.50	0.50	0.50	0.51	0.52	0.49	0.50	0.51	0.51
roberta-large	0.50	0.51	0.49	0.51	0.50	0.51	0.50	0.49	0.50	0.46	0.51	0.48

Table 4: F1 (top) and accuracy (bottom) results on WiC-TSV. The Vanilla columns refer to instances where we run inference with a classifier trained on WIKITIDE directly, without adapting inputs or further fine-tuning. GPT3.5 columns denote a use case where we use GPT3.5 for generating a definition of the target word in the first sentence of the dataset instance, and then run inference on this updated input.

Moreover, we report results reported in previous works to further contextualize the results we obtain, which, to reiterate, are from an unsupervised model not directly optimized for this task. Breit et al. (2021) reports the *all true* baseline on the test split has having Accuracy of 50.8% and F1 of 67.3%. Additionally, they obtain Accuracy scores of of 54.4% and F1 scores of 26.2% with an unsupervised BERT-based model, whereas they find significant improvements (Accuracy, 76.0% and F1-score, 78.8%) for a supervised GBERT-based model. We also find in the work by Zervakis et al. (2022), where they propose target sense verification as an analogy detection task, that they achieve Accuracy scores of 78.6% and F1 of 79.7% on the test set (for supervised approaches), and Accuracy of 61.2% (and 51.3% F1) for an unsupervised approach.

The results of our experiment display the ability of the models before and after bootstrapping on all three sets (train,deveopment and test). The bootstrapped approach considerably increases the Macro-F1 performance of the models with respect to WiC-TSV's Task 1 unsupervised setting baselines (Breit et al., 2021). The results also suggest that while BERT shines on a few occasions, the RoBERTa family of models show the highest performance, with RoBERTa-large bootstrapped being the best with F1 score of 0.53. The vanilla versions of the models perform within a range between 0.30 to 0.34. The bootstrapped versions outperform their non-bootstrapped counterparts in all three datasets, with respect to F1 score. We also observe that the difference in F1 scores between before and after bootstrapped versions can go as high as 17 points, which signify that the models learns better during the bootstrapping process. Finally, we also find that the larger models with more parameters outperform their distilled counterparts in most of the dataset versions.

6 Conclusion

We propose a dataset and methodology to design a classifier for detecting temporal changes in temporal definition pairs. We use weak supervision technique by boostrapping the model an unlabelled dataset in output controlled setting. We also see that bootstrapping a model improves the accuracy of the model as well as makes the model more robust. However, the process requires more time to bootstrap the model and the success of the process depends on the initial training. Although the process has its own limitations, we conclude that the idea of using a classifier to detect information changes in with respect to temporality and training it with boostrapping can result in easement of defining which information is relevant to update a model's knowledge base and can help to miti-

gate the issues that a language model suffers due to temporal misalignment.

Ethics and Broader Statement

This paper is concerned with the automatic construction of a dataset by combining publicly available information in the web. Therefore, it might be possible that incorrect or harmful information is present in this derived dataset, although we welcome efforts by the community to contribute mitigating these risks. The dataset construction process did not involve humans.

Potential risks in the dataset might also include incorrectly flagging new knowledge about any article, as our data source Wikipedia is a publicly editable data source. Therefore the possibility of having conflicting or incorrect information also increases. However, the difference of information, which our classifier is trained to detect can help to detect such outliers and provide some insights about it.

References

Oshin Agarwal and Ani Nenkova. 2021. Temporal effects on pre-trained models for language processing tasks. *arXiv preprint arXiv:2111.12790*.

Tal August, Katharina Reinecke, and Noah A Smith. 2022. Generating scientific definitions with controllable complexity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8298–8317.

Hosein Azarbonyad, Zubair Afzal, and George Tsatsaronis. 2023. Generating topic pages for scientific concepts using scientific publications. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part II*, pages 341–349. Springer.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Anna Breit, Artem Revenko, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. WiC-TSV: An evaluation benchmark for target sense verification of words in context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1635–1645, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual pre-training mitigates forgetting in language and vision.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Marco Del Tredici, Raquel Fernández, and Gemma Boleda. 2018. Short-term meaning shift: A distributional exploration. *arXiv preprint arXiv:1809.03169*.

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *Transactions of the Association for Computational Linguistics*, 3:529–543.

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.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Luis Espinosa-Anke, Horacio Saggion, and Francesco Ronzano. 2015. Weakly supervised definition extraction. In *Recent Advances in Natural Language Processing*.

Luis Espinosa-Anke, Horacio Saggion, Francesco Ronzano, and Roberto Navigli. 2016. Extasem! extending, taxonomizing and semantifying domain terminologies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Luis Espinosa-Anke and Steven Schockaert. 2018. Syntactically aware neural architectures for definition extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 378–385.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271, Melbourne, Australia. Association for Computational Linguistics.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. *arXiv preprint arXiv:2004.14118*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016a. Cultural shift or linguistic drift? comparing two computational measures of semantic change. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2116–2121, Austin, Texas. Association for Computational Linguistics.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016b. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Valentin Hofmann, Janet B Pierrehumbert, and Hinrich Schütze. 2020. Dynamic contextualized word embeddings. *arXiv preprint arXiv:2010.12684*.

Jie Huang, Hanyin Shao, Kevin Chen-Chuan Chang, Jinjun Xiong, and Wen-mei Hwu. 2022. Understanding jargon: Combining extraction and generation for definition modeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3994–4004.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. *arXiv preprint arXiv:2204.14211*.

Mandar Joshi, Kenton Lee, Yi Luan, and Kristina Toutanova. 2020. Contextualized representations using textual encyclopedic knowledge. *arXiv preprint arXiv:2004.12006*.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. Timelms: Diachronic language models from twitter. *arXiv preprint arXiv:2202.03829*.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A Smith. 2021. Time waits for no one! analysis and challenges of temporal misalignment. *arXiv preprint arXiv:2111.07408*.

Timothee Mickus, Denis Paperno, and Matthieu Constant. 2019. Mark my word: A sequence-to-sequence approach to definition modeling. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 1–11, Turku, Finland. Linköping University Electronic Press.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 task 1: Codwoe–comparing dictionaries and word embeddings. *arXiv preprint arXiv:2205.13858*.

Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1318–1327.

Miles Osborne, Ashwin Lall, and Benjamin Van Durme. 2014. Exponential reservoir sampling for streaming language models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 687–692.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Guy D Rosin and Kira Radinsky. 2022. Temporal attention for language models. *arXiv preprint arXiv:2202.02093*.

Maja Rudolph and David Blei. 2018. Dynamic embeddings for language evolution. In *Proceedings of the 2018 World Wide Web Conference*, pages 1003–1011.

Maja Rudolph, Francisco Ruiz, Stephan Mandt, and David Blei. 2016. Exponential family embeddings. *Advances in Neural Information Processing Systems*, 29.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Sasha Spala, Nicholas Miller, Franck Dernoncourt, and Carl Dockhorn. 2020. Semeval-2020 task 6: Definition extraction from free text with the deft corpus. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 336–345.

Sasha Spala, Nicholas A Miller, Yiming Yang, Franck Dernoncourt, and Carl Dockhorn. 2019. Deft: A corpus for definition extraction in free-and semi-structured text. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 124–131.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Amir Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Nguyen. 2020. A joint model for definition extraction with syntactic connection and semantic consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9098–9105.

Hongyuan Xu, Yunong Chen, Zichen Liu, Yanlong Wen, and Xiaojie Yuan. 2022. Taxoprompt: A prompt-based generation method with taxonomic context for self-supervised taxonomy expansion. In *Proceedings of the IJCAI Conference on Artificial Intelligence*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

Dani Yogatama, Chong Wang, Bryan R Routledge, Noah A Smith, and Eric P Xing. 2014. Dynamic language models for streaming text. *Transactions of the Association for Computational Linguistics*, 2:181–192.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2021. Dict-bert: Enhancing language model pre-training with dictionary. *arXiv preprint arXiv:2110.06490*.

Georgios Zervakis, Emmanuel Vincent, Miguel Couceiro, Marc Schoenauer, and Esteban Marquer. 2022. An analogy based approach for solving target sense verification. In *NLPIR 2022 - 6th International Conference on Natural Language Processing and Information Retrieval*, Bangkok, Thailand.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

Ruimin Zhu, Thanapon Noraset, Alisa Liu, Wenxin Jiang, and Doug Downey. 2019. Multi-sense definition modeling using word sense decompositions.