Towards *Content Expiry Date* Determination: Predicting Validity Periods of Sentences

Axel Almquist[1⋆] and Adam Jatowt[2][0000−0001−7235−0665]

[1] SentiSum, London, UK
axel.almquist@sentisum.com
[2] Kyoto University, 606-8501 Kyoto, Japan
adam@dl.kuis.kyoto-u.ac.jp

Abstract. Knowing how long text content will remain valid can be useful in many cases such as supporting the creation of documents to prolong their usefulness, improving document retrieval or enhancing credibility estimation. In this paper we introduce a novel research task of forecasting *content's validity period*. Given an input sentence the task is to approximately determine until when the information stated in the content will remain valid. We propose machine learning approaches equipped with NLP and statistical features that can successfully work on a relatively small number of annotated data.

Keywords: Content validity scope estimation · Text Classification · Natural Language Processing · Machine Learning.

1 Introduction

Estimating validity and outdatedness of information is paramount in the pursuit of knowledge, something that we humans, luckily, are good at. If we stumble upon a month-old news article stating *"Trump is visiting Sweden"* we would be fairly certain that this information would no longer be true. Yet, facing a sentence such as *"Stefan Löfven is the prime minister of Sweden"* we would most likely think the contrary. It is knowledge of the world that permits us to make such judgments: we know that a presidential visit only lasts for a couple of days and that if someone is a prime minister they will probably remain so for a few months or years. Unfortunately, computers generally lack knowledge of this kind and are thus still incapable of making such judgments. In a world where the amount of information is perpetually increasing at a fast rate and the need for correct and valid information is at its peak, this is a problem to be solved.

In this paper, we introduce a novel research task of predicting how long information experessed in natural language content remains valid, a notion that we will refer to as *the validity period of content*. In analogy to product's expiry date, the validity period of content can be used for assessing content's *expiry date*. This would define the approximate time point until which the content

⋆ This work was mainly done at the University of Gothenburg.

can be *"safely consumed"* (i.e., used or published), meaning it should retain its validity until that date. The applications of the proposed task are multiple. Few examples are listed below:

Support for Document Editing: Methods that will flag content with short expected scopes of validity could help with document creation and editing, especially, with documents that are meant to be used over longer time frames.

Fact Checking: Fact checking has become recently increasingly important [6,15,16,22,25,34]. A model for identifying validity periods of sentences could be useful to help recognizing outdated or unreliable facts. Given the current time, the creation time and the predicted validity period of a sentence, one could conclude that the sentence is at risk of being outdated if the current time is outside the predicted validity period.

Enhancing Document Retrieval: Search engines are in a constant state of improvement. Many approaches have tried to make use of temporal information to improve content rankings, often in the form of prioritizing recency [8,12,24,26,28,33]. Validity period could be used to flag outdated content while making the still reliable content rise in ranks. By taking the aggregated validity period of sentences in a document, one could filter or flag documents whose validity scope does not cover the current time.

Maintenance of Collaborative Spaces: A part of the information in large knowledge spaces such as Wikipedia, and sites such as Stack Overflow and Quora, will sooner or later go out-of-date. To keep track of outdatedness, validity period estimation could be used to help flag outdated or soon-to-be outdated content. This content could later be removed or changed appropriately. Validity period estimation could also be used to enhance existing approaches for maintenance of such spaces [14].

Besides introducing a novel research task our goal is to create a model that only uses linguistic and statistical features, and is independent from any domain or knowledge graph. We train machine learning models that given a sentence provide an estimate in the form of a selection over fixed validity periods representing how long that sentence will remain valid. We set up the task to be challenging by accepting content of short length (i.e., a sentence) as input, which means there is often limited or no context available. A sentence-level approach can be especially useful for social network services where the length of messages is typically constrained. The experiments are done on an annotated dataset of sentences extracted from blog posts, Wikipedia and news articles.

In summary, we make the following contributions in this paper:

1. We propose a novel task of predicting the validity period of arbitrary textual content and we discuss its applications as well as future extensions.
2. We train machine learning models to predict the validity periods of sentences given a range of linguistic and statistical features, and analyze their impact.
3. We release a dataset which has a high level of annotator agreement for fostering further research.

2 Related Work

Temporal Information Extraction (T-IE) is concerned with extracting temporal information from text [3]. A large portion of T-IE have been focused on extracting and normalizing temporal expressions, such as "*today*" or "*1995*", a task referred to as temporal tagging — to mention a few temporal taggers: GUTime[3], SUTime[4] [4] and HeidelTime[5]. As temporal expressions are not always present, methods and resources for finding *implicit* temporal cues have been used and developed, including language modeling [19,20], word occurrences statistics [17], word embeddings [9] or TempoWordNet [11].

The importance and usefulness of the T-IE has become increasingly recognized and related tasks have been developed including *focus time estimation* [9,17,23], which is the task of identifying what time the texts refer to, *future-related content summarization* [1,10,18] — the task of collecting or summarizing future related information expressed in text, *text date estimation* [5,19,20] which is about detecting the creation time of text, and *temporal scoping of facts*. As for the last one, systems such as T-Yago [31], CoTS [30], PRAVDA [32], TIE [21], and approaches developed by Gupta and Berberich [2] and Sil and Cucerzan [27] have been developed to give facts temporal scopes. Most of these works rely on the existence of temporal expressions in the context of the facts, i.e. that the fact is expressed along with temporal information (T-Yago, PRAVDA, IE, Gupta and Berberich, Sil and Cucerzan). Other approaches rely on occurrence-based statistics of facts to identify temporal scopes (CoTS). For example, if *"Trump is president of the USA"* starts to occur more often than *"Obama is President of the USA"*, this would be an indication of the end of the temporal scope of the fact in the latter sentence. Some approaches, e.g. T-Yago [31], are mainly aimed on Wikipedia infoboxes and lists instead of on free text, and some only focus on a certain type of facts such as relational facts, e.g. *"X was married to Y"* [27].

Due to the above-mentioned limitations, the previous methods are incapable of dealing with either information that is stated in the absence of any temporal expression or with non-factual information, such as *"I am leaving the office now and I will soon be home."* These limitations may not be of serious concern for the above-listed approaches as they mainly focus on major facts regarding past or scheduled events and states. Hence, these approaches are not applicable on many other types of information (i.e., future and often minor actions and events).

Lastly, while our approach would not extract facts and define exact scopes, we should keep in mind that determining exact scopes is generally impossible for many ongoing or future actions and events, especially, if such events or actions lack any predefined period (cf. eating dinner vs. presidential term).

The closest work to ours is research by Takemura and Tajima [29], who classify tweets into lifetime durations, which are used to decide the urgency of Twitter messages. Their objective is to develop an approach for improving the

[3] http://www.timeml.org/tarsqi/modules/gutime/index.html
[4] https://nlp.stanford.edu/software/sutime.html
[5] https://github.com/HeidelTime/heideltime

flow of tweets by taking into account when messages go out of date, hence prioritizing tweets with short life-lengths and ignoring outdated messages. Although Takemura and Tajima try to predict message's lifetime duration, they focus on Twitter messages rather than arbitrary texts. The authors also only use classes of rather short scope, from minutes to weeks, as they want their classification of urgency to be useful for Twitter. Furthermore, and most importantly, Takemura and Tajima's method relies on non-linguistic features, of which many are rather specific to Twitter (e.g., presence of URLs and a user type which is based on the user's previous messages, frequency of their replies, follow relationships and such), rendering their approach less useful on data outside the platform. Our approach does not share this limitation and is meant to be applicable on any text.

3 Method

3.1 Problem Definition & Setting

The validity period of a sentence in our task is a measure of how long the information in that sentence remains valid after it has been expressed. More formally, we define it as follows:

Definition 1. *Given a sentence s created at time t_s, its validity period is the maximum length of time after which the information expressed in s still remains valid.*[6]

While the above definition is general, we use the following validity periods in this work: *few hours, few days, few weeks, few months* and *few years or more*. The granularity of these scopes is unequal ranging from fine-grained (hours) to more coarse (years), which resembles forward-looking logarithmic timeline representation[7]. This is a more natural way for humans to refer to the future, where the uncertainty increases along with the time span extension. Also note that while we could try to pose the problem as a regression task, the simplification of the prediction to a multi-class problem reduces the complexity of the prediction. Besides, given the cost of data annotation and inherent difficulty even for humans to pinpoint the exact validity range, relying on few fixed classes is a more natural choice.

Formally, our model takes as the input a sentence $s_i = \langle w_1, w_2, ..., w_{|s_i|} \rangle$ where w_j denotes a word and $|s_i|$ is the sentence's length, and outputs validity period y_i of a sentence, $y_i = \{$ *few hours*, *few days*, *few weeks*, *few months*, *few years or more* $\}$[8]. To simplify the computation, we assume that the sentence is created during the assessment time[9].

[6] We assume that content is valid at its creation time.

[7] `https://en.wikipedia.org/wiki/Logarithmic_timeline`

[8] In Experiments in Sec 5, we also test the case with the reduced set of three classes.

[9] Determining the approximate expiry date requires then extending the actual creation time of a sentence with its predicted validity period.

3.2 Feature Engineering

In this section we motivate and explain the modeling of feature groups we use.

LSA: Certain words are bound to be more related to some temporal spans than others. It is fairly intuitive that words such as *"election"*, *"economy"* or *"investigate"* would occur more often in sentences with rather long validity periods than with short ones, and vice versa for words such as *"moment"*, *"walk"* or *"dinner"*. We use Gensim[10] to build a TF-IDF model based on Wikipedia[11]. Based on a vocabulary with over 600k words we create an LSA model using T-SVD (Truncated Singular Value Decomposition) to identify such lexical trends as mentioned above. We reduce the dimensions down to 200 which means that each sentence is represented by the top 200 trends identified using T-SVD.

Average Word Length: The intuition here is that more complicated sentences with longer words might tend to have a longer period of validity. For example, sentences about species, statistics, economics or science in general, in contrast to sentences referring to day-to-day things. While it might not always be the case, this feature might still capture some useful shallow patterns.

Sentence Length: Similar to the average word length, the sentence length may be a sign of the validity period. Longer sentences could be characterized by a longer validity or the vice versa.

POS-Tags: This feature is meant to elicit grammatical patterns throughout the classes. Each sentence in represented by a vector of counts for each POS-tag.

Temporal Expressions: Temporal expressions, if present in text, may serve as explicit markers for when the information expressed in a sentence ceases to be valid. CoreNLP's Name Entity Recognition parser identifies four different types of temporal expressions: DATE, TIME, DURATION and SET. We discard the SET type due to the ambiguous nature of SET expressions[12], their less frequent occurrence and difficulty to be mapped into time granularities. DATE, TIME and DURATION expressions are converted by CoreNLP into Timex expressions, which are normalizations of temporal expressions. These are then converted into one of the eight following time granularities we have chosen to use which give us a generalized representation of temporal expressions in the sense of which time granularity they are related to:

[year, month, week, day, hour, minute, second, now]

The conversion, as exemplified in Tab. 1, is done by using Regex to find the time measure of finest granularity mentioned in the Timex expression. Looking at the first row in Tab. 1 we can see that the finest granularity that is mentioned in the Timex expression is referring to a day, thus the time granularity for that expression will be *day*. The sentences are represented by a vector of counts for each granularity for each time type.

[10] `https://radimrehurek.com/gensim/`

[11] Text dump from 2018-05-01.

[12] TIME, DATE and DURATION expressions often point to a specific point (or duration) in time which means that they can be used as explicit markers for when information ceases to be valid. However, SET expressions, such as "*every day*", does not.

Table 1: Conversion of example temporal expressions (in bold) to time granularities.

Sentence	Time Type	Timex	Time Granularity
"*Today is **the 9th of July***"	DATE	2018-07-09	day
"*It is **12.15** and she is still not here*"	TIME	2018-07-09T12.15	minute
"*I am going away for **a few months***"	DURATION	PXM	month

Sentence Embedding: The meaning of a sentence is naturally an important marker for its validity duration. A sentence about the geographical location of a town has a widely different meaning than a person reporting that they will soon be going to bed. To catch this difference in meaning throughout the classes we created sentence embeddings from the average word embeddings[13] of a sentence.

TempoWordNet: TempoWordNet is an extension to WordNet[14]. TempoWordNet gives information about how WordNet senses are related to the past, present, future or if they are a-temporal. To retrieve this information, we need to know the senses of the words in a sentence. We use a naive disambiguation approach and pick the most frequent sense for each word. Each sentence is represented by the probability for past, present, future and a-temporal which is the average probability across all the words, e.g., the past probability for the sentence is the average probability for past for all words in the sentence.

Lexical Categories: In addition to LSA we use a set of manually picked and validated lexical categories which were extracted from modern fiction. The objective is to capture more informal themes which are representative for day-to-day life-related situations. These lexical categories are provided by Empath [13][15]. Empath generates lexical categories by creating category specific term lexicons from a vector space model using cosine similarity. These lexicons then are manually pruned through crowd validation. Empath identifies categories in a piece of text by looking at which lexicons the occurring terms belong to. We use Empath's 194 pre-validated categories, of which 10 are exemplified below.

[help, office, dance, money, wedding, domestic_work, sleep, ...]

Each sentence is represented by a vector containing normalized scores for all categories.

Global Temporal Associations (GTA): Next we propose a method for finding and representing temporal properties of words and combinations of words. The intuition is that certain words and their combinations such as *("build", "house")* or *("kick", "ball")* presuppose temporal aspects. For example, building a house is something long-lasting while kicking a ball is not. The idea is that by looking at the time granularities of temporal expressions associated with a word or a combination of words one might find their underlying temporal properties. For example, the combination *("build", "house")* should have a stronger association with temporal expression of a granularity of *year* rather than *hour*, unlike, *("kick", "ball")* or *("kick")*.

[13] We used pre-trained word embeddings by Google created based on news: `https://code.google.com/archive/p/word2vec/`

[14] `https://wordnet.princeton.edu/`

[15] `https://github.com/Ejhfast/empath-client`

Table 2: Statistics about sentences extracted from Common Crawl.

Sentences	Sentences with temp exp	Temp exp	Temp exp as verb modifiers
8,800,000	1,811,608	2,249,309	823,944

We calculate Global Temporal Associations (GTA) of content elements based on statistical approach over large scale data to discover the global co-occurrences of time granularities with words and their combinations. For this, we use Common Crawl dataset [16] which is a web dump composed of billions of websites with plain text versions available. To handle noise, a few filtering conditions are used, such as allowing only Latin characters and punctuation-ended lines. After filtering, slightly less than 9 million sentences were parsed.

For each sentence found in the Common Crawl dataset we identify DATE, TIME and DURATION expressions. We only use temporal expressions that are modifiers of a verb. These are identified by looking at the sentence dependency relations. The choice of only using temporal expressions that are verb modifiers is because the meaning of the sentence heavily relies on verbs which often dictate the temporal properties of the sentence. For each extracted verb we find the related subject and object. This results in obtaining SVO combinations related to each temporal expression. The temporal expressions are then converted into one-hot-vectors representing time granularities and added to the count vector for DATE, TIME or DURATION for the verbs, nouns and SVO combinations related to the temporal expressions. These count vectors are stored and updated throughout the extraction process.

It is important to note that we store information about a noun in a subject position separate from information about the same noun in an object position. This is meant to catch potential differences in temporal properties of when a noun occurs as an object or as a subject. Other statistics are also collected to exemplify the dataset and to calculate the GTA, such as word counts, corpus size, numbers of temporal expressions that are of type DURATION, DATE and TIME, and how many of these are verb modifiers, and so on. Looking at Tab. 2 we can see that out of the almost 9 million sentences extracted, only 823k sentences included a temporal expression that is a verb modifier.

Having collected the co-occurence statistics of words and their combinations with temporal granularites as described above, we can proceed to estimating temporal associations in target sentences. For a given sentence from our dataset we extract the subject (S), object (O) and root verb (V) along with the SVO combinations (i.e., SV, VO, SVO). Every word and word combination found in the analyzed sentence is queried for in the count data which contain all the time granularity count vectors. If we can not find a word or combinations we use the count vectors of the most similar word or combination, with the assumption that they share similar temporal properties. To find the most similar word or combination of words we use the concatenation of word embeddings and cosine similarity measure. When we find the time granularity count vectors for a word or a combination of words we calculate the PMI (Pointwise Mutual Information) of

[16] http://commoncrawl.org/

that word or word combination with each level of time granularity. For example, PMI(*('build','house')*, DAY_DURATION) gives the association strength between the term combination *('build','house')* and the duration of *day* granularity.

The GTA features for a target sentence are the PMI vectors for all the six possible words and combinations related to the sentence's root verb (S, V, O, SV, VO, SVO) along with the similarity scores for each. If a certain grammatic position or combination is not found in a sentence, the vectors for that position or combination are filled with zeros.

3.3 Feature Normalization and Selection

After all the feature groups are prepared, each sentence is represented by 907 features. We normalize all features using L2-norm and scale them to fit to [0,1].

To avoid overfitting and to improve efficiency, a feature selection method - Recursive Feature Elimination (RFE) - is used. RFE recursively removes features to obtain the best possible model relative to a machine learning algorithm. After initial testing we found it beneficial to reduce feature count down to 100.

4 Evaluation

In this section, we report experimental results starting with dataset preparation and experiment settings.

4.1 Dataset Construction

As this is a novel task, we needed to create a dataset[17]. This comprised of two challenges; selecting sentences and manually annotating them into the five classes of validity periods. To cover a broad linguistic variations and build a model reflecting the real world, sentences were randomly extracted from three different datasets: blog posts[18], news articles[19] and Wikipedia articles[20]. These different datasets use different types of language, cover different type information, and can be more related to certain time spans than others. Blogs tend to be about day-to-day things, while Wikipedia articles tend to be about general and objective concepts. News are somewhat in between and are often more formal and objective than blogs.

During the extraction, a set of conditions were used to improve the quality of sentences. The conditions removed sentences starting with certain words such as "*and*" and "*this*", ones that were too short or too long based on the character count, ones containing past or future tense verbs, and non-English sentences. Furthermore, we tried to equalize the numbers of sentences with and without temporal expression to maintain balance between sentences with explicit and implicit temporal properties.

The annotation consisted of categorizing sentences into one of the five temporal classes. Sentences that could not be understood or for which a validity

[17] https://github.com/AxlAlm/ValidityPeriods-dataset
[18] http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm
[19] https://www.kaggle.com/snapcrack/all-the-news/data
[20] http://kopiwiki.dsd.sztaki.hu/

Table 3: Example sentences for each class taken from the dataset.

Few hours	*So Michi, Audrey, Joel and myself are all hanging out in Lindas basement.*
Few days	*School starts at a later time on Wednesday but thats no big deal.*
Few weeks	*I am taking a course on learning how to use the program 3d studio max.*
Few months	*I am also playing a gig with the new millennium string orchestra at the beginning of next month.*
Few years or more	*The middle eastern nation of Israel is planning to expand its settlements, its housing areas in the west bank.*

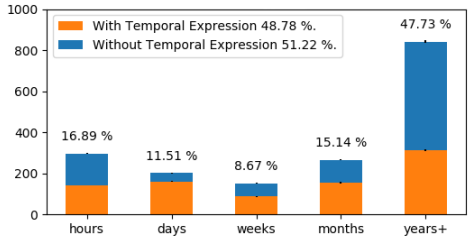

Fig. 1: Distribution of classes and sentences with and without temporal expressions for the original five classes.

period could not be estimated were discarded. Annotators were asked to assume that every sentence was true and created at the annotation time and to estimate when the information stated in each sentence ceases to be valid. All sentences that did not have 100% agreement between two annotators were removed.

The final dataset consists of 1,762 sentences. Tab. 2 contains example sentences for each class, while Fig. 1 describes the distribution of classes along with the portion of sentences with and without temporal expressions.

4.2 Experimental Setting

When extracting sentences for the dataset, Tokenization and POS-Tagging were performed using NLTK[21] and Temporal Tagging by CoreNLP's[22] Name Entity Recognition parser. For feature modeling all previously mentioned NLP methods plus dependency parsing were done with CoreNLP. Normalization was done in the form of lowercasing and abbreviation resolving[23]. For certain features[24] stopwords were also removed.

As the main metric, we use F1-micro to account for the class imbalance. To get robust results and reduce overfitting we use 5-fold cross-validation. T-test was performed on each model using all features in references to all the baselines to mark any significant differences. We contrast the results of LinearSVC[25], SVC_RBF[26], RandomForest[27], KNN[28] and MLP (Multi-Layered Perceptron) to

[21] `https://www.nltk.org/`
[22] `https://stanfordnlp.github.io/CoreNLP/`
[23] e.g. "i'm to "I" and "am".
[24] LSA, average word length, sentence length, POS-tags, Sentence embeddings, TempoWordNet.
[25] For LinearSVC we use $C = 0.7$.
[26] For RBF we use $C = 60$.
[27] We use 150 trees.
[28] Five neighbors are used together with distance weighting.

Table 4: F1-micro using all sentences with original classes. Statistical significance in relation to baselines in each previous box is marked with **.

Models	F1-micro
Random	19.61
Majority Class	47.76
RNN	59.49
MLP (LSA)	39.17**
KNN (LSA)	56.95**
RandomForest (LSA)	60.01**
SVC_RBF(LSA)	61.77**
LinearSVC(LSA)	62.39**
MLP (all features)	53.76**
KNN (all features)	60.07**
RandomForest (all features)	62.75**
SVC_RBF (all features)	67.44**
LinearSVC (all features)	**68.69****

Table 5: F1-micro using all sentences with reduced classes. Statistical significance in relation to baselines in each previous box is marked with **.

Models	F1-micro
Random	34.94
Majority Class	50.14
RNN	70.51
MLP (LSA)	63.61**
KNN (LSA)	61.77**
RandomForest (LSA)	66.15**
SVC_RBF(LSA)	69.48**
LinearSVC(LSA)	70.11**
MLP (all features)	72.50**
KNN (all features)	68.75**
RandomForest (all features)	70.90**
SVC_RBF (all features)	77.37**
LinearSVC (all features)	**78.11****

baseline models that only use LSA as well as to two naive baselines, one that classifies everything randomly and one that classifies every sample with the most common class. MLP uses two dense hidden layers with 500 cells each. Each layer has a 75% dropout and uses Relu activation. The last layer is a dense layer with a Softmax activation and with cells equal to the amount of classes.

We also compare the previously introduced models to RNN where the input is a sentence represented as a sequence of word embeddings. The RNN in constructed by two stacked LSTM layers with 40 cells each preceded by a dense layer with 120 cells. All hidden layers were set to have 75% dropout after tuning and use Tanh activation. The last layer is identical to the last layer of the MLP.

The avoid overfitting the NN's, we stop traning if there are no imporvements after 5 epochs for the RNN and 7 for the MLP. The neural networks are implemented in Keras[29] and for other methods we use the implementations provided by Scikit-learn[30].

Table 6: The impact on the F1-micro when each feature when using LinearSVC is removed for both sets of classes.

features removed	original classes	reduced classes
None	**68.69**	**78.11**
LSA	67.38 ↓	78.33 ↑
avrg word len	67.73 ↓	79.18 ↑
pos-tags	69.14 ↑	79.47 ↑
sent emb	65.29 ↓	74.3 ↓
temp exp	67.44 ↓	77.82 ↓
TempoWordNet	68.63 ↓	79.01 ↑
lexical categories	69.65 ↑	79.24 ↑
GTA	**70.22** ↑	**79.98** ↑

[29] https://keras.io/

[30] http://scikit-learn.org/stable/

Table 7: F1 score of two different models that only use the best features from Tab. 6.

Features	F1 score
LSA, avrg word len, sent len, sent emb, temp exp, TempoWordNet	69.65
sent emb, temp exp	69.03

5 Experimental Results

Tab. 4 shows the main results. We can see that classifiers with all the features outperform the baseline models. Also we see that the NN-based approaches, the RNN and MLP, did not perform well, likely due to the small amount of data. LinearSVC is the most prominent model achieving 68.69% F1-micro.

Next, in Tab. 5 we test how the models perform with reduced classes. The reduction to obtain the three new classes is done in the following way: the *short-term* class is obtained by merging the classes *few hours* and *few days*, *middle-term* is obtained after combining *few weeks* and *few months*, and *long-term* being equal to *few years or more*. The results display similar tendencies observed in the main task: LinearSVC using all features outperforms all other models.

To understand the influence of each feature we create a model using LinearSVC where each feature was omitted. The results are displayed in Tab. 6. Features with a downward-pointing arrow signify loss in F1-micro when being removed. We can see that LinearSVC achieved a F1-micro of 70.22% when GTA was removed. We can also observe that sentence embeddings is the most influential feature.

Based on the information in Tab. 6, additional tests are done with combinations of positively influential features. The results are shown in Tab. 7. Given these results we can conclude that the best observed model uses all features without GTA and achieves a F1-micro of 70.22%.

In the light of the poor performance of GTA, different representations of GTA are tested as shown in Tab. 8 with mixed results. The possible reason for underperformance of GTA is the low co-occurrence of words and their combinations with temporal expressions. For SVO's (Fig. 2) we can see that they mainly co-occurred with zero temporal expressions, and only a minority co-occurred with more than five temporal expressions, a trend that is similar for subject, verbs, objects, VO's and SV's.

Finally, taking a closer look at the classification confusion of our best model (linearSVC with all features except GTA) in Tab. 9, we see that although the model predicts hours and years+ with reasonably good accuracy, it tends to

Table 8: F1 score using different representations of GTA along with other features. First row uses average DURATION PMI vectors, the second uses the weighted average by the cosine similarity score, the third uses the average TIME PMI vectors and the last uses the average DATE PMI vectors.

GTA version	LinearSVC	SVC_RBF	RandomForest	KNN
Average DURATIONN	69.26	68.52	63.58	61.37
Weighted DURATION	70.05	68.23	64.04	62.96
Average TIME	69.54	69.09	64.26	63.3
Average DATE	69.77	68.52	64.95	62.9

Table 9: Confusion probabilities: true class in row and predicted class in column.

Classes	hours	days	weeks	months	years+
hours	**75.17**	6.04	1.68	3.36	13.76
days	19.21	**57.14**	3.45	4.93	15.27
weeks	17.65	9.15	13.07	**30.72**	29.41
months	5.24	6.37	3.0	36.7	**48.69**
years+	1.78	1.31	0.71	3.56	**92.64**

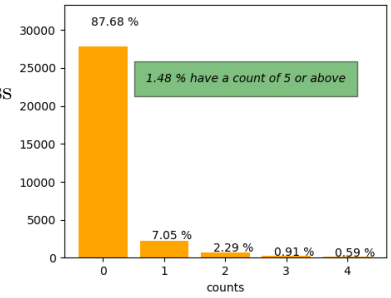

Fig. 2: Five most common co-occurrence counts for SVO and DURATION temporal expressions.

confuse weeks with months and years+. Moreover, we see that months are often confused as years+. This might be due to class imbalance or difficulties annotators had in judging the temporal spans of sentences of these classes.

6 Conclusions & Future Work

The goal of our work is to introduce the task of forecasting the validity period of sentences and to create the first model using only linguistic features. This was motivated by the fact that humans can make such judgments and giving such an ability to computers would help in preventing outdatedness and misinformation.

To be able to design the model, several challenges had to be overcome. The first was to define and represent the task. The second was to create a dataset, which included understanding how sentences should be selected and then annotated to obtain quality data. The second challenge was creating and representing effective features. While sentence embeddings proved to be the most influential features, the core feature, GTA was unfortunately under-performing. Nevertheless, its analysis can provide valuable insights for further improvements. We emphasize that the task is challenging also due to short length of input.

Several further improvements can be proposed. First, as sentence embeddings proved to be the best features, a lot more could be done with their construction and representation. For example, sentence embedding that could reflect the grammatical roles and structure of the sentences [7] might be utilized to further boost the results. Furthermore, including more fine grained validity classes, e.g., classes like "*few minutes*" would make the model more applicable on social media platforms such as Twitter where one might stumble upon sentences such as "*Brad Pitt is standing in front of me in the line at Starbucks!*".

Finally, we propose another task of *extending content's validity period*. It would mean proposing a set of minimal updates of a sentence to move it to a validity class of a longer duration, or to make an invalid sentence valid again while maintaning its semantics. The example applications of this kind of task would be make content of obsolete documents to be useful again or automatically maintaining online content.

Acknowledgments. We thank Nina Tahmasebi for valuable comments and encouragement. This research has been supported by JSPS KAKENHI Grants (#17H01828, #18K19841) and by Microsoft Research Asia 2018 Collaborative Research Grant.

References

1. Baeza-Yates R. Searching the Future. ACM SIGIR Workshop MF/IR 2005
2. Berberich K. and Gupta D. Identifying Time Intervals for Knowledge Graph Facts. Proceeding WWW '18 Companion, In Proceedings of the Web Conference 2018 Pages 37-38 Lyon, France April 23 - 27, 2018.
3. Campos, R., Dias, G., Jorge, A. M., Jatowt, A. (2015). Survey of temporal information retrieval and related applications. ACM Computing Surveys (CSUR), 47(2), 15
4. Chang, A. and Manning, C. SUTIME: A Library for Recognizing and Normalizing TimeExpressions. In Proceedings of the LREC12. Istanbul, Turkey. May 23-25. 2012.
5. Chambers N. Labeling Documents with Timestamps: Learning from their Time Expressions. In ACL '12 Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1. Pages 98-106. Jeju Island, Korea July 08 - 14, 2012
6. Ciampaglia GL, Shiralkar P, Rocha LM, Bollen J, Menczer F and Flammini A. Computational Fact Checking from Knowledge Networks. PLoS ONE 10(6): e0128193. (2015)
7. Clark, Stephen. Vector Space Models of Lexical Meaning. Handbook of Contemporary Semantics. 10.1002/9781118882139.ch16. 2015
8. Dai N., Shokouhi M., and Davison B. D. Learning to Rank for Freshness and Relevance. In Proceedings of the SIGIR11. Beijing, China. July 24-28:ACM Press.95-104. 2011.
9. Das S., Mishra A., Berberich K. And Setty V. Estimating Event Focus Time Using Neural Word Embeddings. In CIKM '17 Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. Pages 2039-2042. Singapore, Singapore November 06 - 10, 2017
10. Dias G., Campos R. and Jorge A. Future Retrieval: What Does the Future Talk About? Workshop on Enriching Information Retrieval of the 34th ACM Annual SIGIR Conference (SIGIR 2011), Jul 2011, Pekin, China. 3 p., 2011.
11. Dias G., Hasanuzzaman M., Ferrari S. and Mathet Y. TempoWordNet for Sentence Time Tagging. 23rd international conference on World wide web companion, Apr 2014, Seoul, South Korea. WWW Companion 14. In Proceedings of the companion publication of the 23rd inter-national conference on World wide web companion, pp.Pages 833-838
12. Efron M. and Golovchinsky G. Estimation Methods for Ranking Recent Information. In Proceedings of the SIGIR11. Beijing, China. July 24-28: ACM Press.495-504. 2011.
13. Fast E., Chen B. and Bernstein M.S. Empath: Understanding Topic Signals in Large-Scale text. In CHI '16 Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems Pages 4647-4657.
14. Grosser Z., Schmidt A. P., Bachl M. and Kunzmann C. Determining the Outdatedness Level of Knowledge in Collaboration Spaces Using A Machine Learning-Based Approach. Professionelles Wissensmanagement. Tagungsband der 9. Konferenz Professionelles Wissensmanagement (Professional Knowledge Management) Karlsruhe, Germany, April 5-7, 2017.

15. Hassan N., Adair B. , Hamilton J. T., Li C., Tremayne M. , Yang J. and Yu C. The Quest to Automate Fact-Checking. In Proceedings of the 2015 Computation + Journalism Symposium.
16. Hassan N., Arslan F., Li C., Tremayne M. Towards Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster. In KDD '17 Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Pages 1803-1812. Halifax, NS, Canada August 13 - 17, 2017
17. Jatowt A. , Yeung C-M. A, Tanaka K. Estimating Document Focus Time. In CIKM '13 Proceedings of the 22nd ACM international conference on Information & Knowledge Management. Pages 2273-2278. San Francisco, California, USA October 27 - November 01, 2013.
18. Jatowt A. and Yeung C-M. A. Extracting Collective Expectations About the Future from Large Text Collections. In CIKM '11 Proceedings of the 20th ACM international conference on Information and knowledge management. Pages 1259-1264. Glasgow, Scotland, UK October 24 - 28, 2011
19. Kanhabua N. and Nrvg. Improving Temporal Language Models for Determining Time of Non-timestamped Documents. In ECDL '08 Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries. Pages 358 370. Aarhus, Denmark September 14 - 19, 2008
20. Kumar, A., Baldridge, J., Lease, M., Ghosh, J. (2012). Dating Texts without Explicit Temporal Cues. CoRR, abs/1211.2290.
21. Ling X and Weld D. Temporal Information Extraction. Twenty-Fourth AAAI Conference on Artificial Intelligence. 2010
22. Lucas Graves. Understanding the Promise and Limits of Automated Fact-Checking. University of Oxford, Reuters institute. 28 Feb 2018 https://reutersinstitute.politics.ox.ac.uk/our-research/understanding-promise-and-limits-automated-fact-checking
23. Morbidoni C., Cucchiarelli A. and Ursino D. Leveraging linked entities to estimate focus time of short texts. In IDEAS 2018 Proceedings of the 22nd International Database Engineering & Applications Symposium. Pages 282-286. Villa San Giovanni, Italy June 18 - 20, 2018
24. Perki J., Buntine W., and Tirri H. 2005. A Temporally Adaptative Content-Based Relevance Ranking Algorithm. In Proceedings of the SIGIR05. Salvador, Brazil. August 15-16: ACM Press.647-648.
25. Popat, Mukherjee, Strtgen, Weikum. Credibility Assessment of Textual Claims on the Web. In CIKM '16 Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. Pages 2173-2178. Indianapolis, Indiana, USA October 24 - 28, 2016.
26. Sato S., Uehar M. and Sakai Y. Temporal Ranking for Fresh Information Retrieval. In Proceeding AsianIR '03 Proceedings of the sixth international workshop on Information retrieval with Asian languages - Volume 11. Pages 116-123 Sapporo, Japan July 07 - 07, 2003.
27. Sil A., Cucerzan S. Temporal Scoping of Relational Facts based on Wikipedia Data. In Proceedings of the Eighteenth Conference on Computational Language Learning, pages 109118, Baltimore, Maryland USA, June 26-27, 2014. 2014 Association for Computational Linguistics.
28. Styskin A., Romanenko F., Vorobyev F., and Serdyukov P. Recency Ranking by Diversification of Result Set. In Proceedings of the CIKM11. Glasgow, Scotland, UK. October 24-28: ACM Press. 1949-1952. 2011.
29. Takemura H. and Tajima K. Tweet Classification Based on Their Lifetime Duration Published in Proc. of CIKM 2012, pp. 2367-2370, Oct. 2012, Maui

30. Talukdar,P. P.,Wijaya,D., and Mitchell,T. Coupled Temporal Scoping of Relational Facts. In Proceedings of the WSDM12. Seattle, USA. February 8-12:ACM Press.73-82. 2012.
31. Wang Y., Zhu M, Qu L, Spaniol M and Weikum G. Timely YAGO: Harvesting, Querying, and Visualizing Temporal Knowledge from Wikipedia. In EDBT '10 Proceedings of the 13th International Conference on Extending Database Technology. Pages 697-700. Lausanne, Switzerland March 22 - 26, 2010
32. Wang Y., Yang B., QU L., Spaniol M and Weikum G. Harvesting facts from textual web sources by constrained label propagation. In CIKM '11 Proceedings of the 20th ACM international conference on Information and knowledge management. Pages 837-846. Glasgow, Scotland, UK October 24 - 28, 2011
33. Yamamoto, Y., Tezuka, T., Jatowt, A., and Tanaka, K. Honto? Search: Estimating Trustworthiness of Web Information by Search Results Aggregation and Temporal Analysis. In Proceedings of the joint 9th Asia-Pacific Web and 8th international conference on Web-age information management conference on Advances in data and Web management. Huang Shan,China. June 16-18.253-264. 2007.
34. You Wu , Pankaj K. Agarwal , Chengkai Li , Jun Yang , Cong Yu. Toward Computational Fact-Checking. In Journal Proceedings of the VLDB Endowment. Volume 7 Issue 7, March 2014 Pages 589-600