Event Annotation and Detection in Kannada-English Code-Mixed Social Media Data

Sumukh S[1], Abhinav Appidi[2], Manish Shrivastava[1]
LTRC, IIIT-Hyderabad[1], PureML[2]
sumukh.s@research.iiit.ac.in, appidiabhinav27@gmail.com,
m.shrivastava@iiit.ac.in

Abstract

Code-mixing (CM) is a frequently observed phenomenon on social media platforms in multilingual societies such as India. While the increase in code-mixed content on these platforms provides good amount of data for studying various aspects of code-mixing, the lack of automated text analysis tools makes such studies difficult. To overcome the same, tools such as language identifiers, Parts-of-Speech (POS) taggers and Named Entity Recognition (NER) for analysing code-mixed data have been developed. One such important tool is Event Detection, an important information retrieval task which can be used to identify critical facts occurring in the vast streams of unstructured text data available. While event detection from text is a hard problem on its own, social media data adds to it with its informal nature, and code-mixed (Kannada-English) data further complicates the problem due to its word-level mixing, lack of structure and incomplete information. In this work, we have tried to address this problem. We have proposed guidelines for the annotation of events in Kannada-English CM data and provided some baselines for the same with careful feature selection.

1 Introduction

With the rising popularity of social media platforms such as Twitter, Facebook and Reddit, the volume of texts on these platforms has also grown significantly. Twitter alone has over 500 million test posts (tweets) per day[1]. India, a country with over 300 million multilingual speakers, has over 23 million users on Twitter as of January 2022[2], and code-switching can be observed heavily on this social media platform (Rijhwani et al., 2017).

Code-switching or code-mixing[3] occurs when "lexical items and/or grammatical features from two languages appear in one sentence"(Muysken, 2000). Multilingual society speakers often tend to switch back and forth between languages when speaking or writing, mostly in informal settings. It is of great interest to linguists because of its relationship with emotional expression (Rudra et al., 2016) and identity. However, research efforts are often hindered by the lack of automated NLP tools to analyse massive amounts of code-mixed data (Rudra et al., 2016).

Below is an example of a code-mixed Kannada-English tweet that has also been translated into English. Named entities have been tagged along with the language tags (*Ka*-Kannada, *En*-English, *NE*-Named Entity, *Univ*-Universal).

> **Ka-En:** Sinchu/*Person/NE* last/*Other/En* month/*Other/En* Kerala/*Location/NE* visit/*Other/En* madidlu/*Other/Ka* #beautiful/*Other/En* :D/*Other/Univ*
>
> **Translation:** *Sinchu visited Kerala last month #beautiful :D*

Event detection in Natural Language Processing (NLP) and Information Retrieval (IR) refers to the process of identifying and extracting relevant information about events from text data. An event can be defined as something that happens at a particular time and place, involving one or more participants and having certain properties or attributes. The emphasis is on detecting the presence of events. This information can be useful for various applications, including news analysis by accurate selection of news messages(Cimiano and

[1]https://www.internetlivestats.com/twitter-statistics/
[2]https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/
[3]The terms "code-mixing" and "code-switching" are used interchangeably by many researchers, and we also use these terms interchangeably

Proceedings of Recent Advances in Natural Language Processing, pages 1007–1014
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_108

Staab, 2004), enhanced risk analytics (Capet et al., 2008), improve traffic monitoring systems (Kamijo et al., 2000), forecasting civil unrest (Ramakrishnan et al., 2014), social media monitoring, event detection, trend analysis, and knowledge graph construction (Ye et al., 2022). Furthermore, by detecting the occurrence of events as early as possible, the performance of risk analysis systems (Capet et al., 2008), traffic monitoring systems (Kamijo et al., 2000) can be improved and forecast civil unrest (Ramakrishnan et al., 2014).

The structure of the paper is as follows. In Section 2, we review the related work. In Section 3, we discuss the annotation methodology and the challenges involved while dealing with ambiguous tokens. In Section 4, we describe the steps involved in corpus creation and data statistics. In Section 5, we describe the baseline systems that have been used. In Section 6, we have discussed the feature selection. In Section 7, we have talked about the experimental setup of our work. In Section 8, we present the results of the experiments conducted. Finally, in section 9, we conclude the paper and discuss the future prospects.

2 Background and Related Work

The study of events dates back quite a long time, and pre-linguistic definitions of events sought to describe and recognise events as "change in aspects of the perceived sense." Before we start with event detection, we should be first clear on what constitutes an event in a sentence. The guidelines for annotation of events have been published in English in 2006 (Saurí et al., 2006), while guidelines for event annotation in monolingual Kannada data was recently published by Prabhu et al., 2020. Automated event mention detection in an open domain setting is a keystone for various information extraction tasks. This task was first brought to light during SemEval-2007, where the shared task *Task 15: TempEval Temporal Relation Identification* (Verhagen et al., 2007) was added as a new task with a focus on identification of temporal constructions. One of the six proposed tasks was concerned with the detection of events mention extent in the text. In early works, most of the methods (Allan et al., 1998, Yang et al., 1998) proposed for event extraction have focused on news articles, which is the only best source of information for current events. More recently, Iqbal et al., 2019 proposed NLP techniques, handwritten rules and WordNet for event extraction from emails. They have used methods like event trigger identification and morphological analysis for event extraction from the email and achieved an accuracy of 72%. With the ability of social media tools to virally popularize news items and their acceptance across the masses, numerous media agencies have been relying on Twitter, Facebook feed pages to disseminate their news highlights. Twitter feeds for Hindi [4][5] and Kannada [6][7] are few examples of social media forums continuously posting the news items. Among the posts made by these feeds, only a small fraction of tweets contain events. Allan et al., 1998 developed the first open-domain event extraction tool (TWICAL) for Twitter data. There have been attempts at event detection from social media streams (Hossny and Mitchell, 2018), but we will not be working on those as part of this work.

We have recently seen an interest related to Kannada-English code mixed data. Sowmya Lakshmi and Shambhavi (2017) have proposed an automatic word-level Language Identification (LID) system for sentences from social media posts. Appidi et al. (2020) reported a work on annotating CM Kannada-English data collected from Twitter and creating POS tags for this corpus. S and Shrivastava (2022) presented an automatic NER of Kannada-English CM data. We are using the dataset created in the above works related to NER and POS tags in our event extraction task.

3 Annotation Methodology

In this section, we shall discuss the method that we have used to annotate our corpus. We label each tokens with the inside-outside-beginning (IOB) format (Ramshaw and Marcus, 1999), where B refers to the beginning of an event, I refers to the token that is part of the event but not the first token and O refers to all other tokens. We propose these principles, which are inspired by TimeML, and are organised by the Part-of-Speech (POS) of the event nugget. Nouns, finite verbs, non-finite verb constructions such as infinitives, and adjectival and adverbial participle constructions are examples of these components of speech. As most of the code-mixed Kannada-English sentences follow the structure of Kannada grammar while swapping language

[4]https://twitter.com/aajtak?lang=en
[5]https://twitter.com/bbchindi?lang=en
[6]https://twitter.com/NewsFirstKan
[7]https://twitter.com/OneindiaKannada

for keys words such as common nouns, we will follow the guidelines that we have proposed for Kannada monolingual event annotation in our paper - Detection and Annotation of Events in Kannada (2020) (Prabhu et al., 2020). For English grammar based sentences, we have the TimeML annotation guidelines (Saurí et al., 2006).

For Kannada grammar based sentences, we have the annotation guidelines from Prabhu et al., 2020. Some examples are given below.

Noun-Nominal events refer to abstract nouns that relate to a temporal phenomenon and inherently convey a notion of finiteness, such as chunavane (election), pasavu (famine), etc.

> **Ka-En:** Karnataka *chunavane* sheeghra agutte antha namme home minister announce madidru
>
> **Transation:** Our home minister announced that Karnataka *election* will be soon

Finite verb- Categorized as events because they denote actions that bring about a change in the state of the world. They possess tense and aspect information, which inherently conveys a notion of temporality.

> **Ka-En:** Prashant avna resignation letter annu *kalisidaane*
>
> **Translation:** Prashant *sent* his resignation letter

Adjectival participle construction, non-finite verb- In Kannada, this involves converting the verb into an adjective to describe the noun involved in the main verb through its previous actions. These constructions exhibit semantics of sequentiality in relation to the main verb and convey a sense of finiteness in the action. The adjectival participle is also inflected with tense, aspect, and modality, indicating its event-like nature.

> **Ka-En:** avnu *oduva* shoes annu wear madida
>
> **Translation:** He wore his *running* shoes

Adverbial participle construction, non-finite verb- Similarly, in Kannada these are used to represent verbs performed by or associated with a noun in the dative or accusative case. Unlike adjectival constructions, there is no direct sequentiality associated with the main verb and the adverbial participle.

> **Ka-En:** *malkondiruva* deer annu Rakesh shoot maadida
>
> **Translation:** Rakesh shot the *sleeping* deer

Infinitives, non-finite verb- In Kannada, these are identified by the characteristic inflective ending of 'lu'. These infinitive forms of the verb are also considered as events in linguistic annotation.

> **Ka-En:** Samiksha eega computer alli *oodalu* hoguttale
>
> **Translation:** Samiksha will now *go read* on computer

Subjunctives, non-finite verb- An uncommon verb form that expresses desires or imagined situations. It is used to indicate events that are uncertain or not guaranteed to happen. As a result, subjunctive verbs are also labeled as events in linguistic annotation. Subjunctives can undergo morphological inflections for tense, aspect, and modality, allowing for the expression of different temporal and modal nuances within the desired or imagined events.

> **Ka-En:** ninage olle aarogya irali anta *bayasutteene*
>
> **Translation:** *I wish* you good health

As stated, we will be using the Inside-Outside-Beginning (IOB) format, so the total number of tags becomes 3 - B, I, and O. An example of the same is given below-

> **Ka-En:** avanu/O Mysore/O alliro/O international/O school/O ge/O hoogi/B science/O odustaane/B
>
> **Translation:** He *goes* to the international school in Mysore and *teaches* science

3.1 Dealing with Ambiguous Tokens

The following are some of the challenges while working with social media data-

- Users tend to use colloquial words/slang on social media and have their own preference of native words. For example, *baralilla* is a Kannada word and it can be written as *brlilla*, *barlilla*, etc.
- Misspelled words are very common on social media. For example, a word like *tonight*

Quantity	Value
Total number of tokens	21,342
Avg. tweet length	9.61
Total tweets	2250

Table 1: Corpus statistics

Event type	Count of occurances
Singe word events	1,955
Multi-word events	1,057

Table 2: Event types statistics

could be written as *tonight, tonite, tonihgt, ton8, etc.,* which posed a significant challenge while building spelling agnostic models.

In case a word has different means in the two languages we are working with and the words refers to an action/event in one language while it does not in the other language, we tag the token with whatever seems appropriate based on the context of the sentence. The annotators shall make such context based decisions for any other ambiguity that they might come across as some words can have multiples meanings, in the same language or across languages.

4 Corpus and Statistics

4.1 Dataset

As we intend to use Part-of-Speech (POS) tags in our work, we have used a subset of the annotated dataset created by Appidi et al., 2020 for POS tagging of Kannada-English code-mixed data. The corpus was created from Twitter[8].

We annotated 2,250 of these code-mixed Kannada-English tweets. For language identification part, we used the tool that was developed by Bhat et al., 2015 but it needed manual checks as social media data differs from the standard language.

The corpus has a total of 21,342 tokens which were tagged for the 7 tags mentioned in the Section 3. The corpus statistics and the event types statistics can be seen in Table 1 and Table 2 respectively.

4.2 Inter Annotator Agreement

Annotation of the dataset for event tags in the tweets was carried out by 2 human annotators having linguistic background and proficiency in both Kannada and English based on the methodology in Section 3. In order to validate the quality of annotation, we calculated the inter annotator agreement (IAA) between the 2 annotation sets of 2,250 code-mixed tweets having 21,342 tokens using Cohen's Kappa (Cohen, 1960) which came up to 0.89 which is fairly high given the challenges we have with the task at hand, discussed in Section 3.1, and the complexity of the annotation guidelines.

Disagreements about the tags were resolved through discussions between the annotators to reach a mutual agreement.

5 Supervised Approaches for Event Detection

Supervised machine learning is a category of machine learning where a model is trained on labeled training data, where each data point has both input features and corresponding output labels. The goal of supervised learning is for the model to learn the mapping between input features and output labels so that it can make accurate predictions on unseen data.

As we have a limited amount of data for our task of event detection (2,250 sentences), we will only explore probabilistic models. They have been described in the following sub-sections.

5.1 Hidden Markov Model

Hidden Markov Models (HMMs), that was first introduced by Baum and Petrie, 1966, have been used for event detection in NLP, particularly in scenarios where the focus is on sequential data (Zhou and Su, 2002, Kupiec, 1992, Jiampojamarn et al., 2007). HMMs are probabilistic models that involve hidden states and observable outputs, making them suitable for modeling the sequential nature of the text. HMMs are finite stochastic automata consisting of two stochastic processes. The first process is a Markov chain with transition probabilities and hidden states. The second process generates observable emissions based on a state-dependent probability distribution. In our context, the emission probability refers to the likelihood of a token being assigned a B, I, or O tag.

It's important to note that HMMs make the simplifying assumption of the Markov property, which assumes that the current hidden state depends only on the previous state. While this assumption may not always hold in complex NLP tasks, HMMs can still be effective for event detection in certain

[8]http://twitter.com/

scenarios. Overall, HMMs provide a framework for modeling sequential data and can be utilized for event detection in NLP when the focus is on capturing the sequential nature of events in text.

5.2 Conditional Random Fields

Conditional Random Fields (CRFs) are probabilistic models used for sequence labeling tasks, such as named entity recognition and event detection in Natural Language Processing. CRFs are trained using optimization techniques such as maximum likelihood estimation (MLE) to estimate the model parameters. The parameters are learned to maximize the log-likelihood of the training data.

CRF is effective for event detection and sequence labeling tasks because it captures dependencies between adjacent labels, considering the contextual information from neighboring words. It enables global optimization by finding the most likely label sequence that maximizes the joint probability, leading to more coherent and accurate predictions. CRF's ability to model the structure of the sequence enhances its performance in identifying event segments within text.

6 Feature Selection

In "A Semantico-Syntactic Approach to Event-Mention Detection and Extraction In Hindi", by Goud et al., they have used specific features for HMMs and some additional features for CRF. We shall use those features along with an additional feature - language identifier (LID) which will either be Kannada (Ka), English (En), Named Entity (NE) or a Universal (Univ) token.

The list of features picked for the HMM are:

1. Word Identity (WI) : This would be the annotated event tag of the token.
2. Part-of-Speech (POS) : This would be the relevant POS tag of the token.
3. Beginning Of Sentence (BOS) : This would be a binary to mark if a token is the first token of a sentence.
4. Capitalization (C) : This is to identify if the token is capitalisized as capitalisation signifies nouns most of the time if not emphasis.
5. Language Identifier (LID) : This is binary feature that is important for code-mixed data as we need to know which language it belongs to - Kannada or English.

The list of features picked for CRF are the following:

1. Word Identity (WI) : This would be the annotated event tag of the token.
2. Part-of-Speech (POS) : This would be the relevant POS tag of the token.
3. Bi-gram features : Adjacent 2 token feature.
4. Tri-gram features : Adjacent 3 token feature.
5. Beginning Of Sentence (BOS) : This would be a binary to mark if a token is the first token of a sentence.
6. Previous word's POS : This feature would help in context understanding of the present token.
7. Previous word's WI : If the previous token's tag was B, then the present token would either be a I or an O tag. This helps in contextual understanding for a CRF.
8. Next word's POS : Similar to previous token's POS tag, next token's POS tag helps understand the present token better.
9. Next word's WI : Similar to previous token's event tag, next token's event tag helps understand the present token better.
10. Language Identifier (LID) : This is binary feature that is important for code-mixed data as we need to know which language it belongs to - Kannada or English.

7 Experimental Setup

In this section, we conducted a series of experiments to assess the impact of different features and model parameters. Our goal was to understand the effect of individual features that we discussed in Section 6 and explore the influence of various model settings that we discussed in the Section 5.

To achieve this, we performed experiments using different combinations of features and systems with rigorous hyper-paparemeter tuning, for both HMM and CRF using GridSearch. We experimented with using a subset of features together and all features simultaneously.

To evaluate the performance of our classification models, we employed 5-fold cross-validation. This approach helped us validate the models by

partitioning the data into five subsets, training the models on four subsets, and evaluating their performance on the remaining subset. We repeated this process five times, each time using a different subset for evaluation.

For the implementation of the algorithms, we utilized the data handling libraries in Python for HMM and CRF++[9] tool for CRF, which are efficient and user-friendly tools for our tasks.

In terms of data splitting, we allocated 60% of the data for training, 10% for validation, and 30% for testing. This division allowed us to train our models on a substantial portion of the data, validate their performance on a separate subset, and finally assess their generalization ability on a dedicated testing set.

By conducting these experiments, we aimed to gain insights into the impact of different features and model parameters, enabling us to make informed decisions about the best configuration for our classification models. We use precision, recall and F1-score as our evaluation metrics.

8 Results and Analysis

Table 3 captures the performance of our models for our dataset. Our best model is the CRF (window size 3) which achieved a weighted average F1-score of 0.53 compared to 0.39 for HMM. The performance of the HMM is in line with expectations. This limitation can be attributed to their ineffectiveness in capturing contextual details and their reliance on a restricted set of features during training, resulting in sparse information availability.

As CRFs are trained to develop a local model at the sentence level for event identification, as we observe improvements in the accuracy of the CRF's local model, it indicates that the engineered features effectively capture all the available information within the sentence's local structure.

Even then, the results are nowhere near perfect but the purpose of the models was just to provide an exploratory baseline for the dataset created with the annotation guidelines.

We should also note that the grammatical structure of the sentences are not standard, making prediction harder. This gets more difficult with Kannada-English code-mixed data as mixing happens at word-level, mostly for Kannada language prepositions and named entities or English language nouns.

[9]https://taku910.github.io/crfpp/

Model	Precision	Recall	F-1 score
HMM	0.38	0.42	0.39
CRF	0.46	0.63	0.53

Table 3: Evaluation of HMM and CRF models for Event Detection

9 Conclusion

In conclusion, our work on Event Detection for code-mixed Kannada-English social media data has yielded the following:

1. **Annotation Guidelines:** We have provided guidelines for annotating code-mixed Kanglish social media data for event detection.

2. **Annotated Corpus:** We have created a new annotated corpus specifically for code-mixed Kannada-English Event Detection. To the best of our knowledge, this is the first corpus of its kind, providing valuable resources for future research in this domain.

3. **Research Problem:** We have identified and addressed the challenge of event detection in code-mixed Kannada-English data as a research problem. Code-mixed data presents unique linguistic complexities, and our work contributes to advancing event detection and other information extraction techniques in this particular context.

4. **Machine Learning Models:** We conducted experiments using probabilistic machine learning models on our annotated corpus. Specifically, we employed Hidden Markov Model and Conditional Random Fields (CRF) models which could be employed as baseline models for further exploration.

As part of future work, we plan to explore downstream tasks like question answering, which makes use of Event Detection for code-mixed data. The size of the corpus can be increased to include more data from varied topics.

10 Acknowledgements

We would like to thank our annotators for their hard work and dedication. We would also like to

thank the anonymous reviewers for their valuable feedback.

References

James Allan, Ron Papka, and Victor Lavrenko. 1998. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45. ACM.

Abhinav Reddy Appidi, Vamshi Krishna Srirangam, Darsi Suhas, and Manish Shrivastava. 2020. Creation of corpus and analysis in code-mixed Kannada-English social media data for POS tagging. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 101–107, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Leonard E. Baum and Ted Petrie. 1966. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554 – 1563.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

Philippe Capet, Thomas Delavallade, Takuya Nakamura, Agnes Sandor, Cedric Tarsitano, and Stavroula Voyatzi. 2008. A risk assessment system with automatic extraction of event types. In *Intelligent Information Processing IV*, pages 220–229, Boston, MA. Springer US.

Philipp Cimiano and Steffen Staab. 2004. Learning by googling. *SIGKDD Explor. Newsl.*, 6(2):24–33.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37 – 46.

Jaipal Singh Goud, Pranav Goel, Alok Debnath, Suhan Prabhu, and Manish Shrivastava. A semantico-syntactic approach to event-mention detection and extraction in hindi.

Ahmad Hany Hossny and Lewis Mitchell. 2018. Event detection in twitter: A keyword volume approach. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1200–1208.

Kanwal Iqbal, Muhammad Yaseen Khan, Shaukat Wasi, Shumaila Mahboob, and Tafseer Ahmed. 2019. On extraction of event information from social text streams: An unpretentious nlp solution. *International Journal of Computer Network and Information Security*, 19:121–131.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379.

Shunsuke Kamijo, Yasuyuki Matsushita, Katsushi Ikeuchi, and Masao Sakauchi. 2000. Traffic monitoring and accident detection at intersections. *IEEE Trans. Intell. Transp. Syst.*, 1(2):108–118.

Julian Kupiec. 1992. Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language*, 6(3):225–242.

Pieter Muysken. 2000. *Bilingual speech*.

Suhan Prabhu, Ujwal Narayan, Alok Debnath, Sumukh S, and Manish Shrivastava. 2020. Detection and annotation of events in Kannada. In *16th Joint ACL - ISO Workshop on Interoperable Semantic Annotation PROCEEDINGS*, pages 88–93, Marseille. European Language Resources Association.

Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, Chris Kuhlman, Achla Marathe, Liang Zhao, Ting Hua, Feng Chen, Chang-Tien Lu, Bert Huang, Aravind Srinivasan, Khoa Trinh, Lise Getoor, Graham Katz, Andy Doyle, Chris Ackermann, Ilya Zavorin, Jim Ford, Kristen Summers, Youssef Fayed, Jaime Arredondo, Dipak Gupta, and David Mares. 2014. 'beating the news' with embers: Forecasting civil unrest using open source indicators.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural Language Processing Using Very Large Corpora*, pages 157–176. Springer.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on Twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1971–1982, Vancouver, Canada. Association for Computational Linguistics.

Koustav Rudra, Shruti Rijhwani, Rafiya Begum, Kalika Bali, Monojit Choudhury, and Niloy Ganguly. 2016. Understanding language preference for expression of opinion and sentiment: What do Hindi-English speakers do on Twitter? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1131–1141, Austin, Texas. Association for Computational Linguistics.

Sumukh S and Manish Shrivastava. 2022. "kanglish alli names!" named entity recognition for Kannada-English code-mixed social media data. In *Proceedings of the Eighth Workshop on Noisy User-generated*

Text (W-NUT 2022), pages 154–161, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Roser Saurí, Jessica Moszkowicz, Bob Knippen, Rob Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. Timeml annotation guidelines version 1.2.1.

B S Sowmya Lakshmi and B R Shambhavi. 2017. An automatic language identification system for code-mixed english-kannada social media text. In *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pages 1–5.

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

Yiming Yang, Tom Pierce, and Jaime Carbonell. 1998. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36. ACM.

Hongbin Ye, Ningyu Zhang, Hui Chen, and Huajun Chen. 2022. Generative knowledge graph construction: A review. *CoRR*, abs/2210.12714.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480.