Is It Out Yet? Automatic Future Product Releases Extraction from Web Data

Gilad Fuchs* eBay Research / Israel gfuchs@ebay.com

Ido Ben-Shaul* eBay Research / Israel ibenshaul@ebay.com

Matan Mandelbrod eBay Research / Israel mmandelbrod@ebay.com

Abstract

Identifying the release of new products and their predicted demand in advance is highly valuable for E-Commerce marketplaces and retailers. The information of an upcoming product release is used for inventory management, marketing campaigns and pre-order suggestions. Often, the announcement of an upcoming product release is widely available in multiple web pages such as blogs, chats or news articles. However, to the best of our knowledge, an automatic system to extract future product releases from web data has not been presented. In this work we describe an ML-powered multi-stage pipeline to automatically identify future product releases and rank their predicted demand from unstructured pages across the whole web. Our pipeline includes a novel Longformer-based model which uses a global attention mechanism guided by pre-calculated Named Entity Recognition predictions related to product releases. The model training data is based on a new corpus of 30K web pages manually annotated to identify future product releases. We made the dataset openly available at https://doi.org/10.5281/zenodo.6894770.

1 Introduction

E-commerce marketplaces and online retailers are constantly updating their inventory with new products. Given the ever growing number of newly released products and their variety, it is becoming increasingly challenging to keep track of upcoming releases. Further, estimating which products are likely to become trendy and highly demanded is an additional task that becomes more difficult with the growth of online E-commerce. For E-commerce marketplaces, whose inventories often include an extremely large variation of products across thousands of different categories, the task of constantly tracking new product releases becomes presumably unfeasible without the leverage of automatic and scalable solutions. In this paper, we demonstrate how an automatic ML-powered extracting pipeline can identify future product releases in billions of websites consisting of unstructured text and rank their demand with high accuracy. We define a future product release identification as identifying both the product name and either its exact release date or a time range.

Our pipeline includes the following main steps. First, the Common Crawl[1] monthly snapshot data is cleaned to include only text by using the pipeline describe in (Raffel et al., 2020; Xue et al., 2021) code[2]. Specifically, we used the already cleaned dataset - "Colossal Clean Crawled Corpus" (C4) (Raffel et al., 2020) and the multilingual variant of the C4 dataset called mC4 (Xue et al., 2021). The next step is a simple but effective combination of data filtering with manually curated release-related key phrases (e.g. "will be released"). Next, a Named Entity Extraction (NER) model is used to detect possible product names and the corresponding releases dates. This step is followed by an additional filtering of non-product related releases using a novel Longformer-based model (Beltagy et al., 2020) ("text2release") which classifies whether a web text indeed includes a future product release or not. We show that using the NER predictions to decide which tokens should have global attention improves the text2release model performance. Next, a consolidation phase aggregates the evidence collected from multiple websites to rank the most likely release date. Last, a buzz calculation for each product is performed based on counting the times each product appears in different websites. An overview of the entire pipeline can be seen in Figure 1. Experimental analysis shows that our pipeline can identify future product release date, in the range of 30 days, with an accuracy between

[*] These authors contributed equally to this work

[1] http://commoncrawl.org/

[2] https://github.com/google-research/text-to-text-transfer-transformerdataset-preparation

Proceedings of EMNLP 2022 Industry Track, pages 273–281
December 9–11, 2020. ©2022 Association for Computational Linguistics

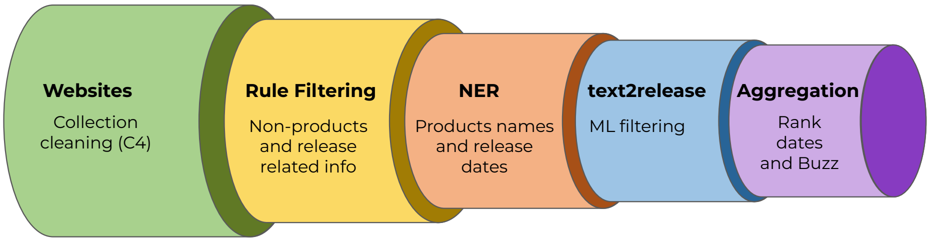

Figure 1: An overview of the product releases identification pipeline.

~70% to ~80%. In addition, our simple buzz calculation shows very high correlation with the actual product demand.

2 Related Work

2.1 Event Detection

There have been several works for predicting events from web data (Zhao, 2020). Some of these focus on discovering local and personal based events for individuals (Foley et al., 2015; Konovalov et al., 2017; Metzler et al., 2012; Li et al., 2017). In (Gravano and Becker, 2011)(Chapter 4), a method for identification of unknown events in social media sites based on trending occurrences is shown. This is done using incremental clustering algorithms, for finding event neighborhoods. Our proposed method is similar in theme to the work done in (Wang et al., 2019), where the aim is to build a database of global events. Other works have proposed to predict global events, mainly through use of data collected from social media platforms (Sakaki et al., 2010; Watanabe et al., 2011; Kim et al., 2018; Farzindar and Khreich, 2015).

In the E-commerce domain, (Yuan and Zhang, 2018) introduce a term frequency–inverse document frequency weighted word embedding to find relevant merchandises for seasonal retail events. However, they rely on a preset marketplace inventory. Finally, (Petrovski et al., 2014) proposes learning regular expressions for attribute extraction of E-commerce Microdata.

2.2 Classifying Long Sequences

The use of Transformers (Vaswani et al., 2017) in NLP applications has become extremely widely used, and accordingly in sequence classification. In general, transformer approaches are often limited to relatively short sequence size. Recently, the Longformer (Beltagy et al., 2020) was introduced to allow using the transformer mechanism on large documents, such as web-pages. Following this work, the Big Bird (Zaheer et al., 2020) model was also proposed to handle the self-attention mechanism on long sequences. In both papers, a combination of the self-attention modifications is shown. In this paper we propose an additional method based on the Global Attention, where the tokens that receive global attention are based on outputs of an NER model. Other works which aim to deal with long sequence sizes have also been presented (Ainslie et al., 2020; Wang et al., 2020; Kitaev et al., 2020).

3 Future Product Releases Identification

3.1 Datasets

The C4 dataset[3] described in (Raffel et al., 2020) was used for the entire product releases identification pipeline development and the ‘text2release’ model training. The C4 dataset is based on Common Crawl’s web data which was released in April 2019. The multi-lingual mC4 dataset (Xue et al., 2021) has 101 languages and is generated from 71 Common Crawl dumps. The product releases identification pipeline was tested over the newest snapshot from August 2020 and only English pages were selected (will be referred from now on as “Aug2020-Eng-mC4”).

3.2 Data Pre-Processing

As each monthly snapshot of the Common Crawl data may include hundreds of millions of web pages, we have decided to use a simple heuristics to select web pages which discuss future product releases. Our approach is based on manually curating a relatively broad set of key phrases which are likely to appear in future product releases web pages. The phrases chosen were general and potentially identify various types of future releases, not necessarily of products. Later stages in the pipeline further enrich the dataset by focusing specifically

[3] https://www.tensorflow.org/datasets/catalog/c4

on product releases. The key phrases and the corresponding number of pages including the specific phrase in the C4 dataset are listed in Appendix A.1, Table 6.

Although the C4 dataset is based on a snapshot from April 2019 of the Common Crawl corpus, the snapshot contains multiple pages from previous years which include future product releases that had already taken place. As our end goal is to detect on a monthly basis future product releases from the latest snapshot released by Common Crawl, we focused our methodology in identifying only future product release that occur after each snapshot release date. Specifically, we added a simple filter, on top of the previously described key phrases, requiring that the text explicitly includes a year string. The C4 dataset has been filtered to a subset containing the string "2019". The Aug2020-Eng-mC4 dataset has been filtered for both the "2020" and "2021", to identify products which were expected to be released at the end of 2020 or the at start of 2021. Although the year filtering may remove potentially relevant web pages that do not explicitly mention the year of release, it makes the dataset more relevant for the specific use-case aimed to be addressed by the pipeline. Following the key phrases and year filtering, the C4 and the Aug2020-Eng-mC4 datasets consist of ~292K and ~305K web pages, respectively.

Next, a subset of web pages were excluded based on a manually curated list of exclusion phrases which were identified to be dominant within releases-related texts and do not have applicable usage for our use case (e.g. mobile applications are usually not sold in E-commerce marketplaces). The main themes of the exclusion list phrases are related to mobile applications, music, TV, films and cars. Last, as manual probing of very long web pages revealed that those web pages rarely discuss future product release, and to ease the pipeline downstream processes, only web pages with text size shorter than 5000 characters were kept, which resulted in keeping ~75% of the web pages. Overall, following the pre-processing steps ~74K and ~78K web pages were selected from the C4 and Aug2020-Eng-mC4 datasets, respectively.

3.3 Entity Recognition of Products and Dates

In order to identify future product release it is essential to detect both the product name and its release date. While for some of the products the new release might consist of only a new model of an existing product, often new releases are for entirely new products. For identifying a release of a new model for an existing product some heuristics can be used (e.g. looking for a pattern of a known product name + variation of a model number). For a previously unseen product such methods are not relevant. Hence, an NER model, capable of identifying product names based on the text context, was used. More specifically, we used the document level NER model FLERT (Schweter and Akbik, 2020), available as part of the Flair package (Akbik et al., 2019). FLERT was trained on the OntoNotes dataset (Weischedel et al., 2013), which includes 18 entity classes, and leverages document-level features by passing a sentence with its surrounding context. In our work we used the *PRODUCT*, *WORK OF ART* and *DATE* entities, as they were identified to be potentially relevant for our use case. Notably, the *WORK OF ART* entity was found to excel in identifying new books and video games specifically. The entity *DATE* was used to identify the different variations of dates described in web pages as free text. Only web pages where the FLERT model predicted the existence of either a *PRODUCT* or *WORK OF ART* were selected for the next step in the future product release identification pipeline. This additional filtering results in exclusion of approximately 50% and 43% of web pages in the C4 and Aug2020-Eng-mC4 datasets, respectively.

3.4 Future Product Releases Classifier

While the FLERT NER model predictions capture which pages include product entities, it can not assure that indeed the web page contains a description of a future product release. In addition, there are multiple cases where tokens are mistakenly predicted to be a *PRODUCT* or *WORK OF ART* entities. To further improve the identification of the web pages specifically describing future product releases, we created a new annotated dataset which includes approximately 30K web pages tagged by crowd-sourced labelers [4]. The 30K web pages were randomly sampled from the releases-enriched C4 dataset (following the steps described at Sections 3.2 and 3.3). Each page was labeled by 4 to 6 annotators, and the labelers were asked to select "text includes future product release" (~63% of pages) or "text doesn't include a future product release"

[4] https://doi.org/10.5281/zenodo.6894770

(~37% of pages). Detaied description of the product releases dataset can be found in Appendix A.2.

In order to improve the identification of future product releases the annotated dataset was leveraged to train a classifier which detects texts mentioning a future product release ('text2release'). As common modern text classification models (e.g. BERT (Devlin et al., 2018)) are limited to 512 sub tokens, and web pages are often significantly longer, we leveraged the pre-trained Longformer model which is capable of handling up to 4096 sub tokens. The tagged data was used to fine-tune the Longformer model, where only web pages having labeling confidence above 0.7 were used (~19,000 web pages). For validation and model testing, only pages with labeling confidence of 1 (i.e. all annotators agreed on the label) were used (~4,700 web pages).

While the original Longformer model uses for classification tasks a global attention in the first token only (specifically, the special 'CLS' token), we examined an alternative architecture which we coin "LongforNER" where global attention is assigned based on NER predicted entities. For this dataset, we chose *WORK OF ART* and *PRODUCT* entities from the FLERT model predictions. The assumption is that greater attention should be given to the product related text in order to better classify if a web page is about a future product release. In Table 1 we compare the test performance of the proposed model (LongforNER) with the results of the vanilla Longformer where global attention are assigned to the CLS token. The NER guided attention resulted in improved performance. It has been shown (Zaheer et al., 2020) that adding random global attention may assist during training to classify long texts. We therefore examined the impact of assigning randomly global attention to a subset of the tokens instead at the specific NER entities (see 'Random' in Table 1), to control the possibility that the improvement of the LongforNER performance is merely due to greater percentage of tokens with a global attention. We confirmed that the percentage of tokens which were assigned randomly with global attention was approximately the same as in the LongforNER version. The LongforNER version also showed better performance comparing to randomly assigned global attention. All models were trained for 30 epochs, with a batch size of 4 and a learning rate of $1 \cdot e^{-6}$, with cosine LR schedule and a minimum value of $5 \cdot e^{-8}$. An AdamW (Loshchilov and Hutter, 2019) optimizer was used in all experiments. The text2release classifier predictions were used to further select web pages of higher probability to include future product release.

Table 1: Comparing predicting future product releases performance metrics while assigning global attention in CLS ('Longformer'), randomly ('Random') or based on NER predictions ('LongforNER'). Each result is an average of 5 different random seed initialization. For the metrics that are based on a given threshold, we use the Youden Index (Youden, 1950). PR-AUC stands for Precision-Recall Area Under the Curve.

Metric	Longformer	Random	LongforNER
PR-AUC	0.8852	0.8834	**0.8901**
F1	0.7926	0.8059	**0.8151**
Accuracy	0.7284	0.7405	**0.7481**

3.4.1 LongforNER Sequence Classification

To further test the advantage of the LongforNER architecture, we test it on the Hyperpartisan news detection dataset (Kiesel et al., 2019). Similar to (Beltagy et al., 2020), we focused on the 'byarticle' dataset, as it's labels are of higher quality. The Hyperpartisan classification task is to decide whether a news article follows a hyperpartisan argumentation, i.e., whether it exhibits blind, prejudiced, or unreasoning allegiance to one party, cause, or person. The Hyperpartisan dataset was previously used to evaluate long texts classifiers (Beltagy et al., 2020; Zaheer et al., 2020). Intuitively, this dataset should benefit from global attention at named entities such as person or organization, as often news, and the hyperpartisan argumentation specifically, involves such entities (e.g. "President Trump and Republicans in Congress must act now to stop new Obamacare taxes..."). Hence, we use the flair[5] 4 classes NER model which identifies the following entities: *PER* (person), *LOC* (location), *ORG* (organization), and *MISC* (other). In Table 2, we show the results of the LongforNER vs. the vanilla Longformer model, using the train/val/test given in (Beltagy et al., 2020). We found the split used in this work to be of particularly high performance. The authors show results on a single split using five different initializations, using the same train/val/test split. Hence, we also measured the performance following splitting the dataset with 5 random splits, using 5 different seed initializations for each. As done in

[5] https://huggingface.co/flair/ner-english-large

Table 2: Average Test F1 on 80/10/10 train/val/test split of the the HyperPartisan dataset using 5 different seed initialization. The split was done either as given in (Beltagy et al., 2020) ('hyper-orig-split') or 5 times randomly ('hyper-new-split').

Dataset	Longformer	Random	LongforNER
Hyper-orig-split	0.9350	0.9243	**0.9390**
Hyper-new-split	0.7638	0.7445	**0.7822**

Section 3.4, the LongforNER performance was also compared to a version where the global attention was assigned randomly. Overall, the LongforNER version shows better performance compared to the CLS-based global attention (Vanilla) and randomly assigned global attention (Random) for both types of the splits. All models were trained for 15 epochs, with a batch size of 4 and a learning rate of $2.5 \cdot e^{-5}$, with linear LR schedule. An AdamW (Loshchilov and Hutter, 2019) optimizer was used in all experiments.

3.5 Pipeline Consolidation

As one of the final goals of the pipeline is to identify the future product release date or a time range, it is necessary to convert the free text describing the date (identified by the NER model) to a structured date format. Specifically, we converted a single date point to a 'DD/MM/YYYY' format and a date range was converted to a tuple of (MIN(DATE), MAX(DATE)). A default of day=15 was used in cases where the release date includes only month and year without a specified day. While the simple patterns of free text dates were found to be parsed successfully with the open source package dateparser[6], for more complicated patterns, which were found to be common in future releases texts, a custom parser was developed. The identified patterns used by the custom parser are summarized in Appendix A.3, Table 8.

As each web page might include several product names and dates, it is essential to link each product name to the corresponding release date. We employ a simple heuristic where we collect all pairs of identified *PRODUCT* or *WORK OF ART* with every *DATE* entity which appear in the same sentence. While this approach does not guarantee that the identified date is indeed the correct release, manual evaluation of sample candidates showed that this is often the case. Moreover, as pairs are collected from multiple websites, aggregating the different dates per product reduces the noise by selecting the most frequent date per product. Any *PRODUCT* and *WORK OF ART* entities which did not have a *DATE* entity in the same sentence were filtered out.

Table 3: Example of 10 identified products ('Product Name'), the suggested release date ('Suggested Date'), and the number of supporting data points for the suggested date ('Date Count').

Product Name	Suggested Date	Date count
assassins creed valhalla	17/11/2020	164
far cry 6	18/02/2021	101
flight simulator	18/08/2020	81
cyberpunk 2077	19/11/2020	70
xbox series x	15/11/2020	66
wwe 2k battlegrounds	18/09/2020	66
kingdoms of amalur	08/09/2020	60
fifa 21	09/10/2020	58
nba 2k21	04/09/2020	51
watch dogs legion	29/10/2020	50

Intuitively, the number of different websites discussing an upcoming product release should be at least partly correlated with the product demand upon its release. In order to count the number of web pages mentioning each product it is possible to count the mentioning of the specific identified product names across all the web pages of a Common Crawl snapshot. However, such a naive approach would yield a large number of false positives, as some product names are not specific enough. For example, searching for the video game "Control" in a full snapshot results in millions of websites. Therefore, we count only web pages where the NER model identified the text as a product name. We refer the number of web pages mentioning the product name as a 'buzz' calculation.

4 Experiments and Results

In order to test the product releases identification pipeline we used the Aug2020-Eng-mC4 dataset for evaluation, described in Section 3.1. Of note, the Aug2020-Eng-mC4 dataset was not used during any stage of the pipeline development, and mimics a case of fetching a new monthly snapshot from Common Crawl to identify future product releases. Running the product releases identification pipeline, as described in Section 1, on the Aug2020-Eng-mC4 dataset resulted in 243 overall products for which the release date was identified. Example of 10 identified products, can be seen in Table 3. Interestingly, 9 out of the top 10 identified prod-

[6] https://github.com/scrapinghub/dateparser

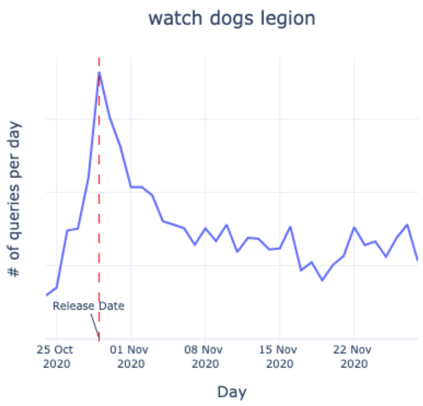

Figure 2: Search query logs per day for the video game 'watch dogs legion' during the 30 days following the game launch.

Table 4: The % of products which their true release date was exactly as the identified release date (P0), within the range of 10 days (P10) or within the range of 30 days (P30) of the true release date.

	P0	P10	P30	N
All	52.7	63.2	69.5	220
Video Games	62.7	71.8	78.2	142

Table 5: Pearson and Spearman's correlation between each product buzz calculation and the total number of search queries in 30 days since the product launch.

	Pearson	Spearman	N
All	0.873	0.647	83
Video games	0.923	0.788	56

ucts are video games. We next manually annotated the categories of the 243 products. The number of identified products from each category are listed in Appendix A.3, Table 9. Overall, the pipeline results in enrichment of video games.

Next, the accuracy of the suggested release dates was evaluated. The true release date was manually labeled per product. Since most of the products were categorized as 'Video Games', 'Smartphones', 'Electronics' and 'Books', the manual labeling of the true release date was done only for the products belonging to these categories. We measure the percentage of products for which the suggested release date was identical to the true release date ('P0') and within the range of 10 or 30 days ('P10', 'P30'). Of note, as in some cases only the expected month and year of the future product release date are mentioned in the text (e.g. "will be released in March, 2021"), this results in lower P0 and P10 compared to P30. Table 4 shows the accuracy of the suggested release dates for all products belonging to the top 4 categories (All) and for the largest category 'Video Games' specifically. The results show that an automatic ML-powered pipeline can identify the release date of more than 200 previously unknown products from a single month web-data snapshot with a error range of 30 days in approximately 70% accuracy. For the largest category of 'Video Games', which on average includes more supportive web pages per product release date compared to the rest of the categories (~10 vs ~4), the P30 accuracy is 78%.

Next, we examine if our buzz calculation can be used to predict at least partly future demand. In order to estimate the product demand we used eBay's search query logs. For each product, that had an actual release date during 2020 (but not before the Common Crawl snapshot release date of 16-Aug, 2020), the demand was estimated by the number of relevant queries found within 30 days since the release date. It is worth noting that not all products were found to be sold on eBay specifically. For simplicity, only products with a dominant single relevant query were examined. Figure 2 shows example of a release date correctly identified by the proposed method, and the query logs in the days after the release. Next, the correlation between the buzz calculation and the demand was calculated. As can be seen in Table 5, a high correlation was found between the buzz calculation and the actual demand, and even higher for products of the largest category of 'Video Games'.

5 Conclusions

In this work we demonstrate the capability of automatically identifying future product releases and their ranked demand, from the free monthly snapshot of the Common Crawl data. The ability to identify product releases in advance is a powerful tool which can be leveraged for multiple downstream applications such as better management of inventory or price updates of outdated models. We also suggest a new NER-guided global attention mechanism to improve long text classification tasks. Last, we release a new dataset consisting of web pages labeled as whether the text includes future product releases or not.

References

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Kenneth Fisher, Philip Pham, Anirudh Ravula, Sumit K. Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers. In *EMNLP*.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Atefeh Farzindar and Wael Khreich. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31:132 – 164.

John Foley, Michael Bendersky, and Vanja Josifovski. 2015. Learning to extract local events from the web. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Luis Gravano and Hila Becker. 2011. Identification and characterization of events in social media.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, D. Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In **SEMEVAL*.

Donghyeon Kim, Jinhyuk Lee, Donghee Choi, Jaehoon Choi, and Jaewoo Kang. 2018. Learning user preferences and understanding calendar contexts for event scheduling. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *ArXiv*, abs/2001.04451.

Alexander Konovalov, Benjamin Strauss, Alan Ritter, and Brendan T. O'Connor. 2017. Learning to extract events from knowledge base revisions. *Proceedings of the 26th International Conference on World Wide Web*.

Cheng Li, Michael Bendersky, Vijay Garg, and Sujith Ravi. 2017. Related event discovery. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Donald Metzler, Congxing Cai, and Eduard H. Hovy. 2012. Structured event retrieval over microblog archives. In *NAACL*.

Petar Petrovski, Volha Bryl, and Christian Bizer. 2014. Learning regular expressions for the extraction of product attributes from e-commerce microdata. In *LD4IE@ISWC*.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW '10*.

Stefan Schweter and Alan Akbik. 2020. Flert: Document-level features for named entity recognition.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Qifan Wang, Bhargav Kanagal, Vijay Garg, and D. Sivakumar. 2019. Constructing a comprehensive events database from the web. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *ArXiv*, abs/2006.04768.

Kazufumi Watanabe, Masanao Ochi, Makoto Okabe, and Rikio Onai. 2011. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. In *CIKM '11*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

W. J. Youden. 1950. Index for rating diagnostic tests. *Cancer*, 3.

Ted Tao Yuan and Zezhong Zhang. 2018. Merchandise recommendation for retail events with word embedding weighted tf-idf and dynamic query expansion. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *ArXiv*, abs/2007.14062.

Liang Zhao. 2020. Event prediction in big data era: A systematic survey. *ArXiv*, abs/2007.09815.

A Appendix

A.1 Data Pre-Processing

Table 6: Key phrases used to enrich releases-related web pages and the number of web pages consisting each key phrase.

Phrase	Number of web pages
"will be released"	556,702
"release date"	537,591
"to be released"	510,501
"will release"	321,214
"product launch"	199,099
"scheduled for release"	45,022
"will launch in"	36,411
"expected to launch"	33,985
"to come out in"	32,793
"release scheduled for"	1,795
Total	2,275,113

A.2 Product Releases Dataset

The product releases dataset is a new annotated dataset which includes approximatly 30,000 web pages tagged by crowd-sourced labelers and openly available at `https://doi.org/10.5281/zenodo.6894770`. The dataset includes sampled web pages from the C4 dataset which were filtered as described in Sections 3.2 and 3.3. In this dataset however, only web pages with text size shorter than 3000 characters were kept (as opposed to 5000). The average number of characters per web page in the dataset is 1469 with a standard deviation of 721 and a median of 1412. The average number of tokens per web page is 244 with a standard deviation of 199 and a median of 236. The web page with the max number of tokens has 646 tokens. The average number of sub-tokens per web-page, using bert-base-uncased WordPiece tokenizer [7], resulted in average of 324 sub-tokens with a standard deviation of 158 and a median of 312 sub-tokens. The web page with the max number of sub-tokens has 1869 sub-tokens. Of note, while the text2release model was trained on the product releases labeled dataset described in Section 3.4, it was also used to generate predictions on longer web pages (up to 5000 characters, as described in Section 3.2). Each page was labeled by 4 to 6 annotators with an average number of annotators of 4.7. The number of annotators per web page can be found in the column “judgments”. The annotators were asked to tag each web page with “text includes future product release” or “text doesn’t include a future product release”. Approximately ~63% of the web pages were found to include a future product release. The annotators were guided to ignore releases that happened in the past, and future releases of non products entities of mobile applications, software, movies and TV shows.

Each web page is associated with a labeling confidence score. The confidence score was calculated by the Appen platform based on the level of agreement between multiple contributors (weighted by the contributors’ trust scores). More details can be found in Appen website [8]. The average confidence score is 0.73 with a standard deviation of 0.146. The median confidence score is 0.743 and the number of web pages with confidence score of 1 is 4,688. Of note, as the annotators were required to label a relatively long and often complicated text, it is expected that not all annotators will agree on each of the label. Examples of positive and negative web pages can be seen in Table 7.

Table 7: Example of web pages labeled as "text includes future product release" (Positive) and "text doesn’t include a future product release" (Negative).

Label	Text
Positive	"gladwell’s previous five books (the tipping point, blink, outliers, what the dog saw, and david and goliath) have all been international bestsellers. in his ground-breaking blink, he explored the role of first impressions in our lives. now he goes deeper, zeroing in on how we make sense of the unfamiliar. talking to strangers will be published september 2019."
Negative	"get involved this spinal health week (20-26 may) to help raise awareness of the importance of being ready for life, so more australians can continue to do the things they love for longer. aca will release weekly blogs in the lead up to spinal health week ... tell us in 50 words or less how chiropractic helps you get ready for life, for your chance to win over $700 worth of prices including a garmin fitness tracker, bose wireless earbuds and a sunbeam stickmaster."

[7] https://huggingface.co/bert-base-uncased

[8] https://success.appen.com/hc/en-us/articles/201855939-How-to-Calculate-a-Confidence-Score

A.3 Pipeline Experiments and Results

The patterns used by the custom date parser can be seen in Table 8. The number of identified products per category can be seen in Table 9.

Table 8: Patterns used to parse date ranges within free text.

Pattern	Example
the [first/second] half of YEAR	the first half of 2021
the [first/last] month of YEAR	the last month of 2019
the [beginning/end] of YEAR	the end of 2020
[early/late] YEAR	early 2021
the [first/.../last] quarter of YEAR	the forth quarter of 2020
[q1/q2/q3/q4] YEAR	q1 2021
MONTH [next/this] year	December this year
the [beginning/end] of MONTH	the end of August
[this/next] SEASON	this summer
the SEASON of [this/next] year	the winter of next year
the SEASON of YEAR	the fall of 2021

Table 9: The number of identified products per category.

Category	Number of Products
Video Games	142
Smartphones	29
Electronics	26
Books	23
Other	23