Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 6170–6179
Marseille, 20-25 June 2022

MAKED: Multi-lingual Automatic Keyword Extraction Dataset

Yash Verma[1], **Anubhav Jangra**[2], **Sriparna Saha**[2], **Adam Jatowt**[3], **Dwaipayan Roy**[1]
[1]Indian Institute of Science Education and Research Kolkata, India
[2]Indian Institute of Technology Patna, India
[3]University of Innsbruck, Austria
{yashv7523, anubhav0603, sriparna.saha, jatowt, dwaipayan.roy}@gmail.com

Abstract

Keyword extraction is an integral task for many downstream problems like clustering, recommendation, search and classification. Development and evaluation of keyword extraction techniques require an exhaustive dataset; however, currently, the community lacks large-scale multi-lingual datasets. In this paper, we present `MAKED`, a large-scale multi-lingual keyword extraction dataset comprising of 540K+ news articles from British Broadcasting Corporation News (BBC News) spanning 20 languages. It is the first keyword extraction dataset for 11 of these 20 languages. The quality of the dataset is examined by experimentation with several baselines. We believe that the proposed dataset will help advance the field of automatic keyword extraction given its size, diversity in terms of languages used, topics covered and time periods as well as its focus on under-studied languages.

Keywords: keyword extraction, text processing, multi-lingual dataset.

1. Introduction

The amount of information being uploaded on internet each day is increasing over time, making it harder to filter out relevant information tailored to an individual. The information over the web spans multiple languages, making the task of multi-lingual keyword extraction quite useful. At the same time the diversity of these languages makes the task also challenging. For example, Le (2015) explains that if we consider Japanese (`ja`) for the task of keyphrase extraction in the legal context, the candidates of interest are words, chunks, and clauses. However, for the same task in English (`en`) language, utilizing similar structural information will lead to a less optimal solution or may not improve the performance since chunks can lead to a noisy output for keyword extraction.

Extracting representative words or phrases from a document is essential to quickly summarize and understand the topics covered within the text. Keyphrases are word(s) that convey the essence or the main topics of the document, and their extraction is essential for supporting or enhancing many downstream tasks in the domain of information retrieval (Medelyan and Witten, 2008), text representation and summarization (Litvak and Last, 2008), document clustering (Han et al., 2007) and so on. Although many scientific articles and news articles are already associated with keywords, most documents are not; hence the development of dedicated models for the keyword extraction task is necessary. A large and diverse annotated corpus will then motivate and foster the development of supervised techniques and evaluation of various keyword extraction methods.

News articles are one of the most consumed and readily available types of documents and have been explored in many state-of-the-art transformer-based models (Zhang et al., 2020; Lewis et al., 2019; Raffel et al., 2019) for tasks like summarization, question generation and answering etc. Keywords are crucial for the news domain and can help in tasks like clustering articles based on keywords, enhancing the search for specific events presented as keywords, and obtaining temporal changes for event recommendation systems.

Previous works introducing datasets for keyword extraction (see Tab. 1. for an overview) rely on small-scale uni-lingual data from the scientific domain. Current deep neural network-based models require however considerable amount of data for training purpose. Yet, to the best of our knowledge, there exist only 3 mono-lingual datasets in the news domain that have up to 500 documents.

Hence, to fill this gap, in this work we propose a multi-lingual keyword extraction dataset from the news domain. It comprises of more than 540K documents and spans over 20 languages: *English*(`en`), *Chinese*(`zh`), *Spanish*(`es`), *Russian*(`ru`), *French* (`fr`), *Ukrainian* (`uk`), *Portuguese* (`pt`), *Japanese* (`jp`), *Tamil* (`ta`), *Hindi* (`hi`), *Marathi* (`mr`), *Gujarati* (`gu`), *Bengali* (`bn`), *Sinhala* (`si`), *Urdu* (`ur`), *Pashto* (`ps`), *Indonesian* (`id`), *Telugu* (`te`), *Punjabi* (`pa`), and *Nepali* (`ne`). This makes it the largest keyword extraction dataset with the highest number of supported languages so far. Note that our dataset also contains images in many documents which have the potential to foster the research in multi-modal keyword extraction.

The major contributions of this work are as follows:

- We release[1] the first ever large-scale multi-lingual keyword extraction dataset covering 20 languages and comprising of 540K+ news articles.
- The performance of various baselines on the proposed dataset, including statistical, graph-based, and supervised keyword extraction methods, is reported.
- It is the largest mono-lingual news keyword extraction dataset for each proposed language, where at least 14 of the covered languages are under-studied and categorized as low-resourced.
- To the best of our knowledge, this is the first cross-lingual keyword extraction dataset for English-Japanese (`en-ja`) pair.

[1]A sample dataset can be found in the project repository at `https://github.com/zenquiorra/MAKED`. The complete dataset will be released in the camera-ready version.

Keyword Extraction Datasets			
Dataset Name	**No. Docs**	**No. Languages**	**Type. Domain**
Krapivin2009	2304	1	PCS
Inspec	2000	1	ACS
wicc	1640	1	PCS
WWW	1330	1	ACS
Schutz2008	1231	1	PCS
cacic	888	1	PCS
Keyword Extraction Datasets in the News Domain			
WikiNews	100	1	NM
110-PT-BN-KP	110	1	NM
500N-KP-Crowd-v1.1	500	1	NM
MAKED (Proposed)	542,134	20	NM

Table 1: Comparison of the proposed dataset with existing keyword extraction datasets. ACS refers to 'Abstracts of Computer Science articles', PCS refers to 'Papers of Computer Science articles', and NM means 'News Miscellaneous'.

2. Related Work

In this section we discuss some existing keyword extraction datasets that are frequently used in the community. A large percentage of these datasets are based on scientific publications since these already contain manually-added keywords.

Krapivin2009: Krapivin et al. (2009) proposed a dataset consisting of 2304 scientific papers from the computer science domain published by ACM. Every article has keyphrases assigned by the authors, and parts of each paper such as abstract and title are separated, enabling extraction based on a given part of an article's text.

Inspec: Hulth (2003) proposed a dataset consisting of 2000 abstracts of scientific journals from the computer science domain, and it has a temporal span from the year 1998 to 2002. For every document, the ground truth keywords are assembled by taking the union of controlled keywords, which are available in the Inspec thesaurus (these may not appear in the document content), and the uncontrolled keywords assigned by the authors.

WWW: Gollapalli and Caragea (2014) proposed a graph-based algorithm CiteTextRank for automatic keyphrase extraction. It utilizes the context in which a document is referred to within a citation network and the content of the document. WWW is constructed as a gold standard annotated dataset to test the performance of CiteTextRank. It has been obtained from the proceedings of the last ten years of the World Wide Web Conference (WWW) and consists of 1330 documents.

Schutz2008: Schutz and others (2008) released a collection of scientific papers collected from PubMed Central, which consists of publications from biomedical literature. The dataset contains 1231 documents; the authors provided the gold keywords for the corresponding documents.

WikiNews: Bougouin et al. (2013) proposed a graph-based algorithm called TopicRank for automatic keyphrase extraction. It relies on the topical representation of the document, such that the vertices in a complete graph are keywords clustered into topics that are ranked using a graph-based ranking model. A French (fr) corpus has been created using the French version of WikiNews that contains 100 news articles with manual annotations added by students.

110-PT-BN-KP: Marujo et al. (2013) proposed a dataset made from 8 TV Broadcast News (BN) programs in Portuguese (pt) language containing 110 news, derived from the European Portuguese ALERT BN database. In-house manual examination produced transcriptions, including punctuation, capitalization, and segmentation. Keyphrases were then manually annotated with the objective to capture keyphrases that summarize each news.

500N-KP-Crowd-v1.1: Marujo et al. (2013) proposed 500N-KP-Crowd-v1.1 consisting of 500 news articles in English (en) language across various categories. Ground truth keywords have been developed through the Amazon Mechanical Turk service, using multiple annotators, and keywords were chosen if these were provided by over 90% of the annotators.

Most of the keyword extraction works have created corpora suitable for testing their proposed keyword extraction methods, and they utilize different parts of text across certain domains. In general, the field lacks a large-scale benchmark dataset that could be used to evaluate existing methods across multiple topics with varying document sizes. Previous works lack the required size to train modern neural-based models or to evaluate the actual performance of unsupervised techniques over a large real-world corpus instead of a small sample space, which may not be representative. The existing datasets are also mono-lingual, and most of them are limited to English language (refer to Appendix A). Since different languages have varying writing styles, there is a need for robust keyword extraction techniques that could handle such variations. The MAKED corpora that we release can act as the benchmark for evaluation, as well as a source for training such robust models. It consists of 20 languages, out of which 9 are among the top 10 most spoken languages globally, and they belong to 5 language families namely *Indo-European*, *Dravidian*, *Austronesian*, *Sino-Tibetan* and *Japanic*. A detailed comparison of our dataset with prior works can be found in Table 1..

3. Dataset

3.1. Dataset Construction

We collect data for 20 languages spanning across different regions of the world. Out of these, English en accounts for 46% of the data. The data is accumulated in a dedicated parser-ready format within a single repository. Major source of data is BBC News[4] articles and links pointing to those articles which have been bootstrapped from corresponding Twitter[5] accounts for the corresponding language. Further links are recursively obtained from bootstrapped articles which also provide hyperlinks to other news articles as references, related works, and suggested articles within the webpage.

BBC News: BBC News is a division of British Broadcasting Corporation responsible for gathering and broadcasting current news affairs. It publishes news across different

[2] https://loc.gov/standards/iso639-2/php/code_list.php

[3] https://loc.gov/standards/iso639-2/php/code_list.php

[4] https://www.bbc.com/news

[5] https://twitter.com/bbc

Lang	#articles	A. tk	Avg. Sent	# Gold Key	A. K. tk	Abs. Go.%
si	2,590	660.80	32.89	9,987	1.65	55.26
pt	4,307	5,809.21	221.61	17,848	1.39	68.16
fr	6,689	753.80	25.70	22,929	1.83	71.07
ja	6,845	1,083.04	34.69	32,346	1.55	48.28
ps	10,140	605.76	20.89	41,757	1.67	55.60
ne	10,933	449.09	27.68	37,334	1.46	47.51
pa	11,364	848.87	39.06	58,841	1.54	53.20
gu	11,682	873.04	50.32	62,485	1.71	57.47
zh	12,926	1,436.83	45.36	60,364	1.90	51.04
bn	13,226	618.85	38.10	45,272	1.32	39.53
id	13,642	907.74	44.64	41,467	1.39	65.67
te	15,061	631.52	52.15	77,430	1.39	62.36
mr	15,736	873.84	63.54	82,331	1.52	56.66
ur	19,835	998.33	1.12	76,481	1.43	42.66
ta	20,835	495.63	33.87	85,830	1.40	61.25
hi	22,286	1,144.29	54.65	95,163	1.56	43.77
uk	25,905	659.61	33.84	83,976	1.26	80.62
es	31,782	3,985.67	127.76	127,237	1.63	67.08
ru	36,654	881.56	36.53	129,075	1.34	81.13
en	249,696	677.96	23.05	730,12	1.72	51.28
Total	542,134	1,219.77	50.37	95,914.05	1.53	57.98

Table 2: **Dataset Statistics.** "A.tk" denotes the average number of tokens in a document for a given language, "Avg. Sent" denotes the average number of sentences in the document for a given language, "# Gold Key" denotes the number of gold keywords in the whole corpus for a language, "A. K. tk" denotes the average number of tokens in given keywords for a language, and "Abs. Go.%" denotes the percent of tokens in keywords absent from the input text.
"Lang" follows language codes defined by the ISO 639-1 standard[3].

regions in various languages. We have selected 20 such languages with different language roots, targeting many regions worldwide and many under-researched languages. Finally, we collected news articles with associated keywords for the corresponding languages.

Obtaining Articles: For every language, we collect links to articles from publicly available corresponding BBC Twitter accounts.

To extend the data, we scrape[6] valid links[7] obtained from the parsed articles for each language. We use Scrapy[8] as our primary tool for crawling news articles and obtaining chunks of data with identification labels[9].

Selecting Keywords: We select articles in BBC News which have keywords associated with them. Further to validate the quality of keywords, we manually verify 100 instances of randomly selected articles against the given keywords in English, Hindi, and Bengali languages[10] confirming their correctness. We assume that this validation results held also for other languages in our corpus due

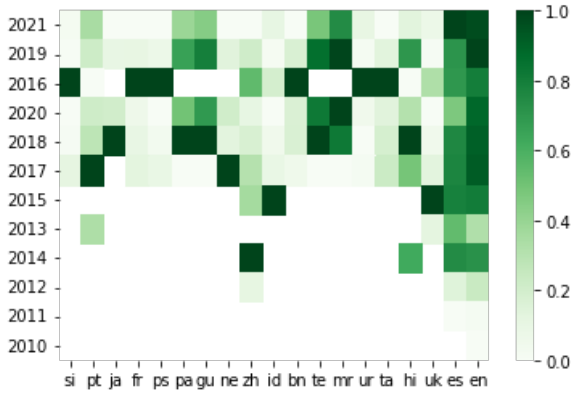

Figure 1: Temporal Span Density for each language in MAKED. Every column represents a distribution of the frequency of articles published for the corresponding year; darker green colors indicate a higher number of published articles in the corresponding year for the language in our corpus. The darkest region represents a score of 1.0, indicating that all corpus articles in the given language were published in the particular year.

to the uniformity across BBC News for all languages.[11]

Ordering Data: These chunks are processed further in Python for ordering and clustering tasks. All chunks are ordered using the identification tags assigned to them during scraping, and elements of articles are accumulated in a hash map based on their identification tags. We design the hash map with tags to optimally obtain specific modalities. The same structure is further written in an JSON file to be ready for use with a dedicated parser designed for our JSON structure.

Final Data: The final dataset consists of text documents in 20 languages saved in JSON format and a parser to access various modalities from each document, with every article having keyword(s) in the corresponding language for that document[12].

3.2. Analysis of Dataset Features

MAKED spans over 20 languages, and contains over 540K documents. Within this corpus, *English* (en) accounts for 250K documents. The dataset contains documents in 6 languages which have more than 20,000 instances, while the smallest corpora consists of 2,590 instances (Sinhala (si) language). We also did a survey on the number of available keyword extraction datasets for each language from various sources including Papers With Code[13], Metatext[14], Kaggle[15] and investigating top results from Web search engines. The survey containing the detailed statistics (no. of speakers, language family, no. of existing datasets) for each language in our corpus can be found in Appendix A.

[6]Data is collected following the terms and conditions mentioned on the website

[7]A link is valid if it points to a BBC article with at least one keyword for the corresponding article.

[8]https://scrapy.org

[9]Identification labels are unique hash values generated using hashlib (https://pypi.org/project/hashlib/) from the URL of an article, which are assigned to every element processed as a separate item from our scrapy implementation

[10]We limit ourselves to 3 languages because of our language understanding

[11]This is also evident from the uniform placement of keywords across various languages a BBC News article webpage

[12]The parser can be found in our project repository at https://github.com/zenquiorra/MAKED

[13]https://www.paperswithcode.com

[14]https://www.metatext.io

[15]https://www.kaggle.com/datasets

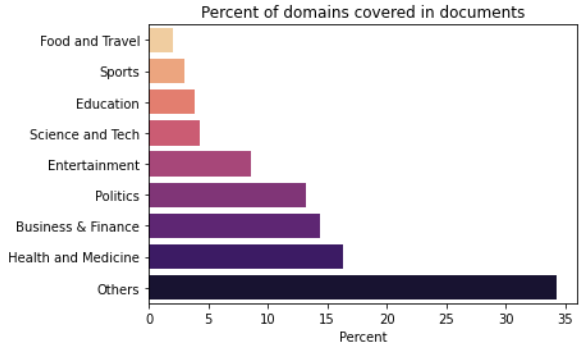

Figure 2: Domains covered in the English (en) corpus of MAKED

MAKED has a temporal span of half a decade for most languages, while some of the languages span for over a decade. Figure 8. shows the yearly density of articles published for every language in our corpus.

To explore the diversity further, we analyze the domains span of our dataset. To investigate the documents' distribution over corresponding domains, we manually annotate a set of randomly chosen 1000 instances from the English (en)[16] corpus. We classify these sample documents into 8 categories:, "Sports", "Business & Finance", "Food & Travel", "Entertainment", "Science & Tech", "Politics", "Health & Medicine" and "Education". We keep an extra category labeled "Others" for domains that do not belong to one of the 8 above categories. The results of this study can be found in Figure 2.

4. Experiments

4.1. Experimental Setup

We do a train-test-validation split with a ratio of 80:10:10 for every language in our dataset. To obtain unbiased results across different languages, we combine publicly available tokenizers and sentence segmenters for multiple languages in a single package[17]. We also define a set of rules for segmentation tasks by analyzing languages that have no such support in external packages[18]. We use segtok[19] for certain Indo-European languages, IndicTokenize[20] for Indian languages, fugashi[21] for Japanese and chinese[22] for Chinese. To analyze our data and run certain baselines, apart from tokenization and segmentation, we also obtain stop words from nltk[23]. For languages not supported in nltk, we collect stop words available in spaCy repository[24].

[16] We restrict ourselves to the English language as it was understandable by all our annotators

[17] The package can be found in the project repository https://github.com/zenquiorra/MAKED

[18] To reduce complexity and size of our implementation, we use only those external packages which offer most functionality.

[19] https://pypi.org/project/segtok/1.1.0/

[20] https://github.com/anoopkunchukuttan/indic_nlp_library

[21] https://pypi.org/project/fugashi/

[22] https://pypi.org/project/chinese/

[23] https://nltk.org

[24] https://github.com/explosion/spaCy/tree/master/spacy/lang

4.2. Baselines

We evaluate the performance of various keyword extraction techniques on our dataset, including two statistical, two graph-based, and one semi-supervised technique. For statistical methods, we employ TF−IDF (Salton and Buckley, 1988) as it is one of the most basic statistical methods to capture the importance of words and is also a basis for many statistics based techniques for keyword extraction. We also evaluate the performance of YAKE (Campos et al., 2020) on our dataset - one of the most recent statistical methods that offers superior performance compared to other techniques. For graph-based approaches, we employ TextRank (Mihalcea and Tarau, 2004) being one of the simplest and most well-known graph-based techniques, often serving as a basis for other graph-oriented techniques. We also explore MultiPartiteRank (Boudin, 2018) as its one of the most recent and better performing graph-based models.

Finally, we explore a semisupervised way to extract keywords using a multilingual embedding and a classification mechanism on top of it. We utilize the pretrained checkpoint mT5_multilingual_XLSum[25](Hasan et al., 2021) of the MT5[26] encoderdecoder model. We supplement this model to obtain keywords by using the KeyBERT (Grootendorst, 2020) package, which uses an embedding model and classifies words from the text document into keywords.

4.2.1. Statistical Approaches

YAKE!: YAKE! (Campos et al., 2020) is an automatic keyword extraction technique that utilizes multiple statistical features from the text to assign scores to words and phrases. It ranks the candidates to obtain keywords for a given text.

TF-IDF: Term frequency inverse-document frequency (Salton and Buckley, 1988), is a statistical measure that determines how important a word is within a document given a collection. TF determines how often a word occurs in a document, and IDF determines the significance of a word, given a corpus. Words with high TF-IDF scores are considered as candidates for keywords.

4.2.2. Graph-based Approaches

TextRank: TextRank (Mihalcea and Tarau, 2004) is a graph-based keyword extraction technique. It utilizes the structure of the text, taking the co-occurrence of words into account to create a graph structure, and then it further determines keyphrases that are the most central to the target document.

MultiPartiteRank (MPR): MultiPartiteRank (Boudin, 2018) is a graph-based keyword extraction technique; it utilizes a multipartite graph structure to represent candidates and topics within a single graph and exploits relationships to improve candidate selection. It determines keyphrase preference using a novel selection mechanism.

4.2.3. Semi-Supervised Approach

We denote the semi-supervised approach we study here as MT5 throughout the paper. In this approach, we utilize a

[25] https://huggingface.co/csebuetnlp/mT5_multilingual_XLSum

[26] https://huggingface.co/docs/transformers/model_doc/mt5

pre-trained model mT5_multilingual_XLSum (Hasan et al., 2021) which is a multi-lingual model pre-trained on the XL-Sum(Hasan et al., 2021) dataset. Note that, this dataset obtained its data from the BBC News[27] domain, same source as ours. We pass this model along with a text document inside the publicly available KeyBERT (Grootendorst, 2020) package which utilizes cosine similarity based on embeddings to obtain keywords.

5. Results and Discussion

5.1. Evaluation Metrics

We use the following metrics to evaluate the performance of tested models:

- We generate a confusion matrix based on predicted keywords and gold keywords based on keyword overlap. We do not segment keywords containing more than one word and consider the actual match to evaluate recall scores. We denote this metric by "em-R" (exact match Recall).
- Levenshtein Distance: We use the Levenshtein distance (Levenshtein and others, 1966) to evaluate edit distance between gold keywords and predicted keywords by taking an average of top-n scores as discussed further. We normalize the Levenshtein Distance between 0 and 1 by dividing the distance by the maximum length of the two compared strings. We then subtract it from 1 to obtain a similarity score; a perfect match implies a Levenshtein distance similarity score of 1. We denote this metric by "lev".
- Jaro-Winkler Distance: We use the metric proposed by Winkler (1990) to evaluate the similarity between two gold and extracted keywords; we employ it similar to the way we did with Levenshtein distance similarity. To compute this similarity, we subtract the distance from 1, to have an even comparison across all metrics such that a perfect match implies a Jaro-Winkler similarity score of 1. We denote this metric by "jar".

We analyze the performance of the baselines on the MAKED dataset using these metrics over the top 5, top 10, and top 15 extracted keywords against the gold keywords.
The first metric relies on the actual match and captures the performance of baselines across all languages; however, for many languages, the ground-truth keywords cannot be verbatim located in the input document. This can be observed for Indic Languages such as Hindi (hi) and Tamil (ta), for which the "em-R" scores (refer to top of Table 3) are considerably lower across all baselines compared to other languages like English (en). Note that we have not implemented stemming for evaluation by the exact match. Extracted keywords can be composed of multiple words depending on the model used. Hence, "em-R" alone is insufficient to evaluate the performance of a baseline across various languages. To capture the semantics, we employ Levenshtein and Jaro-Winkler distance as our metrics. We assume that, since these two metrics rely on the edit distance between two strings, that any two words having a lower edit distance (having similar terms in them[28]) will imply that the words are likely similar and hence semantically close. Jaro-Winkler additionally puts more weight on matching prefix, which further enhances the evaluation of extracted keywords where the representation is different.
In regards to this, employing these similarity metrics, we observe that languages for which "em-R" scores are extremely low can still capture some meaning, as shown in Table 3.

5.2. Top n scores

As we increase the number of extracted keywords and evaluate the exact match, in general, there is a rise in "em-R" scores, sometimes by a large magnitude. This is intuitive since the increase in *n* increases the sample size of extracted keywords to be matched against the gold keywords.
Overall, we do not observe significant change in the Levenshtein scores and Jaro-Winkler scores as we increase the size of *n*. For some cases, we even observe a drop in scores (e.g. the Jaro-Winkler score for TextRank on ja dataset drops from 0.025 to 0.022 when n is changed from 5 to 10 as can be seen in the bottom section of Table 3). This may imply that the corresponding baseline has extracted semantically best possible keywords for smaller n values, but it may still give a better "em-R" score if an exact match is found further. The significantly low MPR scores for some of the languages can be explained by the lack of parts of speech available in the nltk toolkit, which is implemented by many of the packages used for executing our baselines.
Moving from top-5 to top-10 extracted keywords changes the trend for the best performing baseline across different metrics. However, this depends on multiple factors, including the average number of sentences within the corpus for a given language and the number of gold keywords. We further discuss other such features in Section 8..

6. Sources of Errors

We discuss the potential sources of errors in our results in three broad categories:

1. Software induced errors: We utilize various publicly available packages for analysis and technique evaluation. A major part of the experimentation section includes techniques like Tokenization, Segmentation, Parts of Speech tagging, Stop Words filtering, etc. which are implemented in multiple layers within such packages. We have attempted to make the baselines compatible and uniform across languages. However, even these systems are not perfect, and some error could be credited to such technical inadequacies.

2. Author/Editor Bias in articles: The scraped dataset consists of manually written keywords for news articles, and hence some human error could have been inculcated in the process. For example, in the document presented in the case study (refer to Fig. 3), another potential keyword could have been "Brexit", however,

[27] https://www.bbc.com/news

[28] Levenshtein distance between the word "booking" and "book" is 3, while between "booking" and "back" is 5. "Booking" and "book" are semantically closer compared to "back"

Baselines	YAKE!			TF-IDF			TextRank			MPR			MT5		
Languages	Top 5	Top10	Top 15	Top 5	Top10	Top 15	Top 5	Top10	Top 15	Top 5	Top10	Top 15	Top 5	Top10	Top 15
si	**0.002**	**0.002**	**0.002**	**0.002**	**0.002**	**0.002**	0.000	**0.002**	**0.002**	0.000	0.000	0.000	0.000	0.000	0.000
pt	0.123	0.177	**0.208**	0.121	0.177	0.177	0.007	0.016	0.026	0.130	0.193	0.193	0.002	0.005	0.009
fr	0.077	0.121	0.149	0.100	0.133	0.133	0.003	0.009	0.015	0.098	**0.153**	**0.153**	0.003	0.003	0.003
ja	0.145	0.190	**0.211**	0.020	0.029	0.029	0.002	0.012	0.028	0.111	0.147	0.147	0.002	0.004	0.004
ps	0.096	0.142	0.172	0.151	**0.198**	0.198	0.000	0.000	0.000	0.000	0.000	0.000	0.008	0.011	0.015
ne	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.002	**0.004**	0.000	0.000	0.000	0.000	0.000	0.000
pa	0.001	**0.002**	**0.002**	0.001	**0.002**	**0.002**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
gu	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
zh	0.051	0.058	**0.061**	0.038	0.046	0.046	0.010	0.050	0.100	0.036	0.044	0.044	0.001	0.001	0.001
bn	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000
id	0.152	0.223	**0.268**	0.192	0.251	0.251	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.005	0.007
te	0.001	0.001	0.001	0.001	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000
mr	0.001	**0.002**	**0.002**	**0.002**	**0.002**	**0.002**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ur	0.005	0.008	0.009	0.100	**0.162**	**0.162**	0.000	0.000	0.000	0.000	0.000	0.000	0.010	0.014	0.017
ta	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000
hi	**0.004**	**0.004**	**0.004**	0.003	**0.004**	**0.004**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
uk	0.011	0.026	0.039	0.046	**0.069**	**0.069**	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.002	0.003
es	0.098	0.145	**0.175**	0.121	0.156	0.156	0.007	0.018	0.027	0.109	0.154	0.154	0.002	0.005	0.006
ru	0.175	0.326	**0.436**	0.049	0.074	0.074	0.188	0.324	0.418	0.122	0.158	0.185	0.000	0.001	0.001
en	0.086	0.146	0.196	0.157	**0.218**	**0.218**	0.022	0.045	0.066	0.117	0.177	0.177	0.009	0.013	0.017

Baselines	YAKE!			TF-IDF			TextRank			MPR			MT5		
Languages	Top 5	Top10	Top 15	Top 5	Top10	Top 15	Top 5	Top10	Top 15	Top 5	Top10	Top 15	Top 5	Top10	Top 15
si	0.053	0.051	0.050	0.056	0.055	0.055	0.103	**0.107**	**0.107**	0.000	0.000	0.000	0.022	0.023	0.021
pt	**0.167**	0.166	0.163	0.159	0.158	0.158	0.157	0.161	0.163	0.167	0.164	0.164	0.035	0.037	0.038
fr	0.153	0.155	0.154	0.149	0.151	0.151	0.149	0.156	0.157	0.160	**0.162**	**0.162**	0.037	0.039	0.040
ja	**0.056**	0.045	0.038	0.031	0.029	0.029	0.029	0.027	0.024	0.051	0.042	0.042	0.024	0.025	0.022
ps	0.161	0.154	0.149	**0.176**	0.163	0.163	0.137	0.143	0.143	0.000	0.000	0.000	0.035	0.038	0.038
ne	0.031	0.031	0.032	0.032	0.032	0.032	0.088	0.089	**0.090**	0.000	0.000	0.000	0.031	0.032	0.027
pa	0.056	0.052	0.051	0.057	0.058	0.058	0.083	0.088	**0.090**	0.000	0.000	0.000	0.025	0.026	0.021
gu	0.045	0.045	0.045	0.044	0.046	0.046	0.094	0.096	**0.097**	0.000	0.000	0.000	0.024	0.025	0.021
zh	0.020	0.015	0.013	0.036	0.032	0.032	**0.058**	0.056	0.052	0.040	0.032	0.032	0.035	0.035	0.029
bn	0.038	0.038	0.038	0.036	0.037	0.037	0.094	0.096	**0.097**	0.000	0.000	0.000	0.028	0.030	0.024
id	**0.189**	**0.189**	0.187	0.187	0.183	0.183	0.114	0.125	0.131	0.000	0.000	0.000	0.040	0.043	0.043
te	0.031	0.030	0.030	0.032	0.031	0.031	**0.094**	0.100	0.102	0.000	0.000	0.000	0.023	0.025	0.020
mr	0.034	0.027	0.027	0.029	0.030	0.030	0.090	**0.092**	**0.092**	0.000	0.000	0.000	0.024	0.025	0.021
ur	0.135	0.139	**0.140**	0.133	0.134	0.134	0.079	0.088	0.093	0.000	0.000	0.000	0.037	0.039	0.038
ta	0.014	0.014	0.014	0.015	0.014	0.014	0.119	0.124	**0.126**	0.000	0.000	0.000	0.027	0.028	0.023
hi	0.044	0.044	0.044	0.046	0.046	0.046	0.090	0.096	**0.099**	0.000	0.000	0.000	0.027	0.029	0.023
uk	0.122	0.126	0.127	**0.132**	0.131	0.131	0.084	0.086	0.087	0.000	0.000	0.000	0.035	0.036	0.036
es	0.167	0.166	0.165	0.164	0.162	0.162	0.167	0.170	**0.172**	0.167	0.164	0.164	0.036	0.038	0.039
ru	0.014	0.021	0.032	0.124	0.135	**0.164**	0.024	0.032	0.031	0.032	0.045	0.046	0.138	0.141	0.141
en	0.172	**0.173**	0.171	0.172	0.169	0.169	0.166	0.167	0.165	0.167	0.165	0.165	0.037	0.039	0.039

Baselines	YAKE!			TF-IDF			TextRank			MPR			MT5		
Languages	Top 5	Top10	Top 15	Top 5	Top10	Top 15	Top 5	Top10	Top 15	Top 5	Top10	Top 15	Top 5	Top10	Top 15
si	0.022	0.024	0.019	0.024	0.024	0.019	0.023	0.024	0.024	**0.037**	**0.037**	**0.037**	0.255	0.271	0.280
pt	0.036	0.038	0.036	0.035	0.038	0.034	0.035	0.037	0.037	0.036	0.038	0.035	0.467	0.476	**0.479**
fr	0.037	0.039	0.036	0.037	0.039	0.033	0.036	0.038	0.038	0.037	0.039	0.036	0.460	0.472	**0.476**
ja	0.026	0.027	0.022	0.023	0.024	0.020	0.025	0.022	0.020	0.026	0.026	0.022	**0.068**	0.062	0.059
ps	0.035	0.037	0.034	0.037	0.039	0.033	0.033	0.035	0.035	0.051	0.051	0.051	**0.443**	0.443	0.441
ne	0.029	0.032	0.023	0.029	0.032	0.023	0.032	0.032	0.031	0.052	0.052	0.052	0.240	0.246	**0.248**
pa	0.026	0.026	0.019	0.024	0.025	0.018	0.024	0.026	0.025	0.038	0.038	0.038	0.215	0.232	**0.242**
gu	0.023	0.024	0.017	0.023	0.024	0.017	0.024	0.025	0.025	0.037	0.037	0.037	**0.295**	0.294	0.293
zh	0.030	0.030	0.023	0.032	0.034	0.024	0.038	0.034	0.029	0.035	0.034	0.027	**0.059**	0.052	0.050
bn	0.025	0.028	0.020	0.025	0.028	0.020	0.030	0.031	0.030	0.047	0.047	0.047	0.216	0.231	**0.237**
id	0.040	0.043	0.041	0.041	0.044	0.038	0.040	0.042	0.042	0.057	0.057	0.057	0.469	0.480	**0.485**
te	0.022	0.025	0.018	0.022	0.025	0.018	0.025	0.026	0.025	0.039	0.039	0.039	0.222	0.239	**0.246**
mr	0.023	0.024	0.017	0.021	0.022	0.016	0.026	0.026	0.026	0.040	0.040	0.040	0.226	0.230	**0.234**
ur	0.036	0.037	0.036	0.038	0.035	0.024	0.034	0.035	0.034	0.055	0.055	0.055	0.401	**0.407**	**0.407**
ta	0.024	0.025	0.021	0.024	0.025	0.021	0.028	0.029	0.029	0.041	0.041	0.041	0.234	0.254	**0.263**
hi	0.026	0.028	0.020	0.026	0.027	0.020	0.028	0.029	0.029	0.046	0.046	0.046	**0.266**	0.265	0.263
uk	0.034	0.033	0.033	0.034	0.035	0.030	0.036	0.034	0.033	0.051	0.051	0.051	0.410	0.420	**0.424**
es	0.035	0.038	0.037	0.035	0.038	0.033	0.035	0.038	0.038	0.036	0.038	0.036	0.476	0.485	**0.488**
ru	0.035	0.035	0.034	0.386	0.404	0.404	0.037	0.036	0.035	0.047	0.048	0.048	0.437	0.447	**0.450**
en	0.037	0.039	0.038	0.037	0.039	0.033	0.036	0.038	0.039	0.037	0.039	0.036	0.465	0.475	**0.479**

Table 3: Performance of various baselines against the MAKED dataset. Top table presents the exact match recall (em-R) scores, middle table shows the Levenshtein Distance Similarity and bottom table displays the Jaro-Winkler similarity scores. Top 5, Top 10, and Top 15 refer to the Top *n* number of extracted keywords from the corresponding baselines taken into consideration for each evaluation metric.

Intel not considering UK chip factory after Brexit
Pat Gelsinger told the BBC that before the UK left the EU, the country "would have been a site that we would have considered".But he added: "Post-Brexit... we're looking at EU countries and getting support from the EU".Intel wants to boost its output amid a global chip shortage that has hit the supply of cars and other goods.The firm - which is one of the world's largest makers of semiconductors - says the crisis has shown that the US and Europe are too reliant on Asia for its chip-making needs. Intel is investing up to $95bn (£70bn) on opening and upgrading semiconductor plants in Europe over the next 10 years, as well as boosting its US output.But while Mr Gelsinger said the firm "absolutely would have been seeking sites for consideration" in the UK, he said Brexit had changed this."I have no idea whether we would have had a superior site from the UK," he said. "But we now have about 70 proposals for sites across Europe from maybe 10 different countries. "We're hopeful that we'll get to agreement on a site, as well as support from the EU... before the end of this year."Microchips are vital components in millions of products from cars to washing machines, but they have been in short supply this year due to surging demand and supply chain issues. It has led to shortages of popular goods like cars and computers and driven up prices - issues Mr Gelsinger said were set to continue into Christmas."There is some possibility that there may be a few IOUs under the Christmas trees around the world this year," he said. "Just everything is short right now. And even as I and my peers in the industry are working like crazy to catch up, it's going to be a while."He said things would "incrementally" improve next year but were unlikely to stabilise until 2023. **Nobody should be too dependent.** Intel's expansion comes as the overall market for semi-conductors is set more than double in the next seven years to around $800bn. The firm also hopes to secure subsidies from US and European politicians, who feel their reliance on Asia for chips could threaten national security. Today the US only produces around 12% of the world's semiconductors, while Korea's Samsung and Taiwan Semiconductor Manufacturing Company (TSMC) account for 70% of global supply. "It is clearly part of the motivation of a globally balanced supply chain that nobody should be too dependent on somebody else," Mr Gelsinger told the BBC. Intel will continue outsourcing some of its chip-making but eventually hopes to make most of its products in-house. Competing won't be easy, though. Chip-making is still far cheaper in Asia and Intel's rivals continue to expand. TSMC, the world's largest contract maker of semi-conductors, will spend $100bn on increasing capacity over the next three years while Samsung invests $205bn.Mr Gelsinger said he is confident Intel can still regain its leading edge. "This is an industry that we created in the US, Intel's the company that puts silicon into Silicon Valley," he said. "But we realise these are good companies, they're well capitalised, they're investing, they're innovating together. So we have to re-earn that right of unquestioned leadership."
Gold Keywords - 'Companies', 'Intel', 'Semiconductors'
YAKE - 'gelsinger said', 'pat gelsinger told', 'said', 'intel', 'gelsinger', 'pat gelsinger', 'would', 'gelsinger told', 'europe', 'supply', 'said brexit', 'asia', 'year', 'next', 'world', 'chip', 'brexit', 'firm', 'site', 'well'
TF-IDF - 'intel', 'gelsinger', 'mr gelsinger', 'europe', 'asia', 'mr gelsinger said', 'gelsinger said', 'chip', 'gelsinger told', 'bbc'
TextRank - '- issues mr gelsinger', 'supply chain issues', 'chip -', 'global supply', 'supply chain', 'semiconductor manufacturing', 'largest contract', '-', 'mr gelsinger', 'global chip', 'supply', 'leading edge', 'confident intel', 'national security', 'european politicians'
MPR - 'pat gelsinger', 'intel', global chip shortage', 'cars', 'supply', 'bbc', 'world', 'next', 'asia', 'europe'
mT5 - 'europe maybe', 'intel company', 'gelsinger told', '100bn increasing', 'making needs', 'edge industry', 'conductors spend', 'chip shortage', 'largest makers', 'firm hopes', 'world semiconductors', 'brexit looking', 'continue outsourcing', 'uk left', 'national security'

Figure 3: Case Study of an article from the English (en) part of our MAKED dataset. On top the title† and content of the document is provided, followed by the gold keywords (ground-truth) and the baseline extracted keywords. The keywords for baselines are ordered in the descending order of their rank provided from the corresponding frameworks.
† The title of the document is added for reader's reference, it has not been passed to any baseline.

the ground-truth doesn't have it. This can be cause by the writer's or the editor's bias.

3. Abstractive nature of keywords: As illustrated in Table 2, $\sim$ 58% keywords are not verbatim present in the input document. Hence it is a limitation of the current keyword extraction frameworks that are unable to generate keywords. Hence this "abstractiveness" is possibly a major cause for the low evaluation scores, making the task of generating and evaluating keywords even more challenging.

7. Case Study

We describe here a case study (refer to Fig. 3) to provide the readers with a sample from the proposed dataset, and to illustrate the performance of each baseline on a randomly chosen example. From the example we can make the following observations:

- In the current example, the TF−IDF model and MPR model output succinct uni-gram/bi-gram outputs, whereas the other three models give noisy outputs (especially TextRank, which even extracts tri-grams[29]).
- Only TextRank and mT5 are able to extract keywords that contain 'Semiconductors'. However, the extracted bi-grams that contain that word result in 0 of the exact match (em-R) score (this helps explain the poor em−R scores for these models to some extent.)
- YAKE, TF−IDF, and TextRank are able to directly predict the uni-gram 'intel' while mT5 predicts 'intel company'. TextRank also predicts a semantically similar keyword 'confident intel' and places it at the 13th rank.
- TextRank predicts an hyphen '-' as a keyword, illustrating the simplistic nature of the technique.
- YAKE, TF−IDF, and TextRank extract redundant keywords (containing the uni-gram 'gelsinger'). Overall, TF−IDF and MPR seem to extract more natural keywords based on this case study.

8. Correlation

We next analyze the correlation between results of baselines and various dataset features including the per-

[29]Since the average keyword token length for the en dataset is 1.72, a model generating tri-grams would lead to poor exact match scores.

cent of keywords that are not verbatim present in our text ("abs_gold"), the average number of tokens in the keywords ("avg_tok_k"), the average number of sentences("avg_sent") and the average number of tokens in the text ("avg_tok") for all languages.

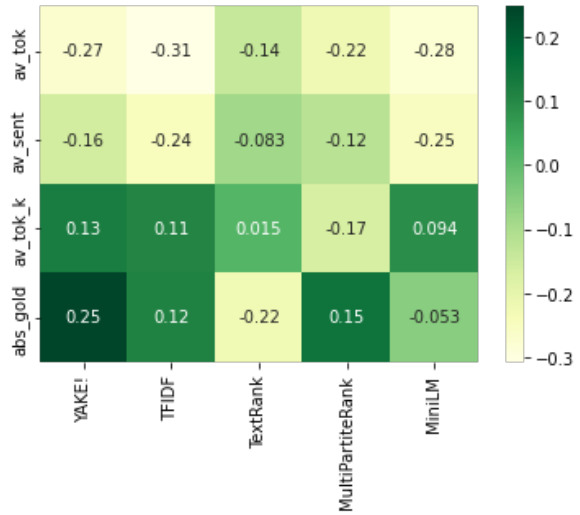

Figure 4: Correlation of various baselines for the top-15 exact match Recall scores with different features

Figure 5: Correlation of various baselines for the top15 Levenshtein scores with different features.

We use the top-15 "em-R" scores and "lev" scores, which refers to exact match recall and Levenshtein distance respectively. The correlation was calculated using the `Pearson` correlation.

For both cases, we observe that all baselines are negatively correlated with the average number of tokens; as the average number of tokens increases, the extraction tasks get difficult, given that the number of "candidate keywords" also increases. This is more prominent in the case of statistical approaches since they are affected by the frequency of terms occurring within a document, as indicated in the Fig. 4 and Fig. 5.

We also observe a performance drop across all five baselines with an increase in the average number of sentences within a document. This drop is even more prevalent for the semantic metrics (Fig. 5) since similar meaning can be conveyed by multiple sentences in larger documents.

We don't observe any significant correlation between the average number of tokens within a keyword. It can also be noted that `TextRank` negatively correlates with the number of absent gold tokens for the exact match recall score (refer to Table 4), while it correlates exceptionally well for the Levenshtein scores. We believe that the tendency of Text Rank to generate bi-grams/tri-grams over uni-grams can explain this phenomenon to some extent. The model implementation only considers co-occurrence among the input document words, making it a simple model (refer to Section 7. for the case study).

9. Dataset Applications

`MAKED` is primarily designed for developing techniques for automatic keyword extraction and technique evaluation. It can be also be used in the following ways:

1. Development of large deep learning frameworks: `MAKED` can be used for training and development of deep neural networks for keyword extraction, utilizing the size of the corpus and the diversity of the topics covered within the news domain.

2. Word Embeddings: Leveraging the size of data available in `MAKED`, the dataset can be used to develop large-scale word embeddings for low resource languages. Since it also captures temporal data of over a decade, creating word embeddings on this data is likely to generalize different writing styles and capture changes in words over time.

3. Parallel Model: The dataset consists of a cross-lingual corpus of Japanese-English (`ja-en`) document pairs. It can then be utilized to create cross-lingual embeddings to further the development of neural networks based cross-lingual keyword extraction.

4. Multi-modal keyword extraction: The dataset can be used to explore and motivate multi-modal keyword extraction. Note that due to the lack of existing works in multi-modal keyword extraction, we cannot provide any baselines for it. However, we plan to explore this as a problem in our future work.

10. Conclusion

In this work, we present the largest multi-lingual keyword extraction dataset in 20 languages, as well as a cross-lingual dataset for a pair of languages obtained from the British Broadcasting Corporation (BBC News). We study the performance of various techniques on our dataset and report their results. In future works, we plan to explore various neural based models for keyword extraction on our dataset with the motivation of developing new word embeddings for low resource languages.

11. Bibliographical References

Bhowmik, R. (2008). Keyword extraction from abstracts and titles. In *IEEE SoutheastCon 2008*, pages 610–617. IEEE.

Boudin, F. (2018). Unsupervised keyphrase extraction with multipartite graphs. *arXiv preprint arXiv:1803.08721*.

Bougouin, A., Boudin, F., and Daille, B. (2013). Topicrank: Graph-based topic ranking for keyphrase extraction. In *International joint conference on natural language processing (IJCNLP)*, pages 543–551.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., and Jatowt, A. (2020). Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Gollapalli, S. D. and Caragea, C. (2014). Extracting keyphrases from research papers using citation networks. In *Twenty-eighth AAAI conference on artificial intelligence*.

Grootendorst, M. (2020). Keybert: Minimal keyword extraction with bert.

Han, J., Kim, T., and Choi, J. (2007). Web document clustering by using automatic keyphrase extraction. In *2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*, pages 56–59. IEEE.

Hasan, T., Bhattacharjee, A., Islam, M. S., Mubasshir, K., Li, Y.-F., Kang, Y.-B., Rahman, M. S., and Shahriyar, R. (2021). XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online, August. Association for Computational Linguistics.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223.

Krapivin, M., Autaeu, A., and Marchese, M. (2009). Large dataset for keyphrases extraction.

Le, T. N. T. (2015). Study on language processing methods for supporting understanding and using multiple legal documents.

Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Litvak, M. and Last, M. (2008). Graph-based keyword extraction for single-document summarization. In *Coling 2008: Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization*, pages 17–24.

Marujo, L., Gershman, A., Carbonell, J., Frederking, R., and Neto, J. P. (2013). Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. *arXiv preprint arXiv:1306.4886*.

Medelyan, O. and Witten, I. H. (2008). Domain-independent automatic keyphrase indexing with small training sets. *Journal of the American Society for Information Science and Technology*, 59(7):1026–1040.

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., and Chi, Y. (2017). Deep keyphrase generation. *arXiv preprint arXiv:1704.06879*.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Schutz, A. T. et al. (2008). Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods. *M. App. Sc Thesis*.

Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Appendix - A (Keyword Extraction dataset Survey)

We perform a literature survey for keyword extraction datasets, covering up 20 languages present in the MAKED dataset. We also collect other relevant information like the total number of speakers across the globe and the Family for each of these language. We used visualcapitalist[30] which has data sourced from Ethnologue[31] to obtain the total number of speakers and the parent in language family tree for each of these 20 languages. We found that there are no existing keyword extraction datasets for 11 out of the 20 languages proposed in our dataset. For example, there is no dedicated keyword extraction dataset for Dravidian languages. The one that exists for Telugu (te) consists of news article snippets where the headlines are considered gold keywords.

Languages			
Language Code	**No. Speakers**	**Family**	**No. Datasets**
English (en)	1,132M	Indo-European	26
Chinese (zh)	1,117M	Sino-Tibetan	1
Hindi (hi)	615M	Indo-European	0
Spanish (es)	534M	Indo-European	2
French (fr)	280M	Indo-European	3
Bengali (bn)	265M	Indo-European	2
Russian(ru)	258M	Indo-European	1
Portuguese (pt)	234M	Indo-European	1
Indonesian (id)	199M	Austronesian	1
Urdu (ur)	170M	Indo-European	0
Japanese (ja)	128M	Japanic	0
Marathi (mr)	95M	Indo-European	0
Telugu (te)	93M	Dravidan	1
Tamil (ta)	81M	Dravidan	0
Gujarati (gu)	61M	Indo-European	0
Ukrainian (pa)	33M	Indo-European	0
Punjabi[32] (pa)	33 M	Indo-European	0
Nepali (ne)	25M	Indo-European	0
Pashto (pa)	21M	Indo-European	0
Sinhala (si)	17M	Indo-European	0

Table 4: Survey of available datasets in comparison with the number of speakers.

For the English (en) language, the only large-scale dataset available belongs to the academic domain (Meng et al., 2017), where abstracts are typically considered as the input text. However, Bhowmik (2008) showed that using just an abstract is not sufficient for keyword extraction, making the existing datasets insufficient either in terms of quality or quantity. The rest of the datasets for keyword extraction in the English language are several magnitudes smaller in scale.

We note that roughly 80% languages covered in MAKED are low resource languages, even though some of them belong to the top-10/20 most spoken languages in the world. On the other hand MAKED also includes some languages which are among the less spoken languages globally and are not yet explored for keyword extraction tasks.

[30] https://www.visualcapitalist.com/100-most-spoken-languages/

[31] Obtained from 22^{nd} edition published in the year 2019 https://www.ethnologue.com/world.