MUKAYESE: Turkish NLP Strikes Back

Anonymous ACL submission

Abstract

Having sufficient resources for a language X lifts it from the *under-resourced* languages class, but does not necessarily lift it from the *under-researched* class. In this paper, we address the problem of the absence of organized benchmarks in the Turkish language. We demonstrate that languages such as Turkish are left behind the State-of-the-Art in NLP applications. As a solution, we present MUKAYESE, a set of NLP benchmarks for the Turkish language that contains several NLP tasks. For each benchmark, we work on one or more datasets and present two or more baselines. Moreover, we present four new benchmarking datasets in Turkish for language modeling, sentence segmentation, and spellchecking and correction.

1 Introduction

Although some human languages, such as Turkish, are not classified as under-resourced languages, only a few research communities are working on them (Joshi et al., 2020). As a result, they are left behind in terms of developing state-of-the-art systems due to a lack of organized benchmarks and baselines. In this study, we aim to fill this gap for the Turkish language with MUKAYESE (Turkish for Benchmarking), an extensive set of datasets and benchmarks for several Turkish NLP tasks.

Having a sufficient amount of resources in a language lifts it from the *under-resourced* class, but it does not necessarily stop it from being *under-researched*. We survey several tasks in Turkish NLP and observe an absence of organized benchmarks and research. We demonstrate how the lack of benchmarks affects languages like Turkish and how it can keep the state of research lag behind the state-of-the-art of NLP. We accomplish this by presenting state-of-the-art baselines that outperform previous work significantly.

In our work on MUKAYESE, we survey seven NLP tasks in the Turkish language, and evaluate available datasets in Turkish for these tasks, and describe the process of creating four new datasets for tasks that do not have public or accessible datasets. Furthermore, we provide at least two baseline models/methods per task besides evaluating existing methods. More details are enlisted in Table 1.

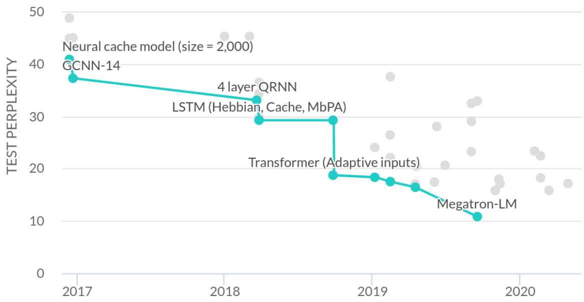

Figure 1: Word perplexity of different models on the test set of WikiText-103 language modeling benchmark dataset[1].

Our overall contribution for Turkish NLP can be summarized as the following: (a) Set of seven organized benchmarks for NLP. (b) Four new datasets in Turkish for language modeling, sentence segmentation, and spellchecking and correction. (c) Dataset splits for fair benchmarking. (d) Several replicable baselines for each task. (e) Benchmarking state-of-the-art methods on Turkish.

The rest of the paper is organized as follows: We give a background on the importance of benchmarking and review similar efforts on NLP in Section 2. We explain the approach we follow for each task in Section 3. We provide dataset statistics, evaluation details, and explain the baselines for each task in 4.

2 Benchmarks and NLP

Following the research on NLP over the years, it can be observed how datasets and benchmarks are very important in measuring the progress of NLP.

For instance, we can observe the progress of

[1] image credits: paperswithcode.com

Task	Datasets	Metrics	Baselines
Language Modeling	- **TrNews-64** - **TrWiki-67**	- Bits-per-char - Perplexity	- Adapt. Trans. - SHA-RNN
Machine Translation	- WMT-16 - MuST-C	- BLEU	- ConvS2S - Transformer - mBART50
Named-Entity Recognition	- WikiANN - Milliyet-NER	- CoNLL F-Measure	- BiLSTM-CRF - BERT - BERT-CRF
Sentence Segmentation	- **TrSeg-41**	- Segment F1-Score	- spaCy - Punkt - Ersatz
Spellchecking & Correction	- **TrSpell-10**	- F1-Score - Accuracy	- Zemberek - Hunspell
Summarization	- MLSUM	- Rouge-L - METEOR	- Transformer - mBART50 - mT5
Text Classification	- OffensEval - *News-Cat*	- F1-Score - METEOR	- BiLSTM - CNN Text - BERT

Table 1: List of the NLP Tasks we work on for the Turkish language in MUKAYESE. We list the datasets, metrics, and baselines we use for each task. New datasets presented in this paper are marked in **bold**, and ones for which we present train/test splits are marked in *italic*.

language modeling for English by looking at the WikiText-103 benchmarking dataset (Merity et al., 2017); See Figure 1.

Likewise, the SQuAD dataset (Rajpurkar et al., 2016) can be used to observe the progress of English Question Answering, and GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019) benchmarks for English Language Understanding.

Such progress has been enabled by the existence of benchmarks, which allowed for fair and meaningful comparison, and showed if there is a room for improvement. In addition, organized benchmarks and datasets allow for the research community to make progress with minimal amount of domain knowledge. This is especially important when it comes to languages with less number of speakers.

This is essential if we want to include other communities in the development of under-resourced and under-researched languages. Since research communities are more likely to contribute when such organized tasks are presented (Martínez-Plumed et al., 2021).

3 Methodology

We focus in MUKAYESE on under-researched tasks of NLP in the Turkish language. We define the required elements for each benchmark as a triplet of (Datasets, Evaluation, Baseline).

After defining the task and assessing its importance, we define its inputs and outputs and construct the following three key elements:

Datasets are the first element to consider when it comes to a benchmark. We define the minimum requirements of a benchmark dataset as follows: accessible with reasonable size and quality and a shareable format.

Unless used in a few-shot setting, benchmarks with small datasets will lack generalizability, and models trained on them might suffer from overfitting. On the other hand, training models on huge datasets might be costly and inefficient.

Another feature to assess is the quality of the dataset. A manually annotated dataset with a low Interannotator Agreement (IAA) rate is not suitable for benchmarking. Moreover, to build a generalizable benchmark, we need to consider using a dataset representing the general domain, e.g., sentence segmentation methods of editorial texts do not work on user-generated content such as social media posts as we show in Subsection 4.4.

Evaluation is the second element of benchmarks. We need to define one or more metrics to evaluate and compare methodologies. We have to answer the following questions before deciding on one metric or more for a benchmark: (a) Is what this metric measures what our task aims to do? (b) How well does it correlate with human judgment? (c) Are there any issues to consider in these metrics? (Using accuracy to measure performance on an unbalanced set does not give a representative idea of model performance).

Baselines are the final element of benchmarking. In order to build a good performance representation of different methodologies, it is better to diversify our baselines as much as possible. For instance, we can compare pretrained vs. non-pretrained approaches, rule-based systems vs. trained systems, or unsupervised vs. supervised models.

4 Tasks

We provide benchmarks in form of (dataset, evaluation, baseline) triplets for each the following NLP tasks.

4.1 Language Modeling

Language modeling is a generative process, which focuses on modeling the probability $P(X)$ of a text sequence of n tokens, where $X = (x_1, x_2, ..., x_n)$, and $P(X) = \prod_{i=1}^{n} P(x_i|x_{<i})$. This type of language modeling is known as Auto-regressive (AR) or causal language modeling. The main objective of the model is to learn to estimate the probability of a given text sequence or a corpus. In our work, we focus on neural approaches for this task (Bengio et al., 2003), where we present two new benchmarking datasets for AR language modeling, and report the results of two different baseline models.

Datasets We present two different datasets for AR modeling, namely **TRNEWS-64** and **TRWIKI-67**, along with their train/validation/test splits (See Table 2). These datasets are presented in a similar fashion to enwik8 (Hutter, 2006) and WikiText (Merity et al., 2017) English datasets.

TRWIKI-67 is a language modeling dataset that contains 67 million words of raw Turkish Wikipedia articles. We extracted this dataset from a recent Turkish Wikipedia dump[2] using WikiExtractor (Attardi, 2015). Additionally, further preprocessing was applied to get rid of the redundant text. Only the articles' raw text and titles were kept and presented in their cased format (with no upper/lower case transformations). For the tokenization process, we train a sentencepiece unigram model (Kudo, 2018) with a vocabulary size of 32K, only using the training split of the dataset. Although we advise using the tokenized version of this dataset to encourage reproducibility, we provide a raw version of this dataset that can be utilized as a benchmark for language modeling tasks on character, subword, or word level.

	#articles	#words	#tokens	avg.sent
TRWIKI-67				
Training	374K	63.5M	4M	12.8
Validation	10K	1.7M	4M	13.3
Test	10K	1.7M	139M	12.9
Total	394K	67M	147M	12.8
TRNEWS-64				
Training	140K	59.7M	421M	23
Validation	5K	2.1M	15M	22.8
Test	5K	2.1M	15M	22.9
Total	150K	64M	450M	23

Table 2: Statistics about TRWIKI-67 and TRNEWS-64 dataset splits. The column *Avg. sents* refers to the mean average number of sentences per article. Tokens are characters for TRNEWS-64 and sentencepiece for TRWIKI-67.

TRNEWS-64 is a character language modeling dataset that contain 64 million words of news columns and articles that was retrieved from TS Timeline Corpus (Sezer, 2017). It can be utilized as a benchmark for modeling long-range dependencies in the Turkish language, as it contains relatively long documents (See Table 2). This dataset consists of a mix of news articles collected from different journals about various domains and topics. Since trnews-64 is intended for language modeling on character level, articles were lightly preprocessed, and no further tokenization was applied.

Evaluation Language models are trained on minimizing the negative log-likelihood (NLL) of the training set, and their performance is measured based on how well they can generalize on the test set:

$$\text{NLL}(X_{test}) = -\frac{1}{n}\sum_{i=1}^{n} log\ p_\theta(x_i|x_{test} < i) \quad (1)$$

Word or sub-word level language models are evaluated using the word perplexity (PPL) metric, a derivative of NLL. On the other hand, character language models are evaluated using entropy-based Bits-per-character (BPC) metric, which is also an-

[2] https://dumps.wikimedia.org/trwiki/20210720/: accessed on 20 July 2021.

other derivative of NLL (Huyen, 2019). We consider PPL for the evaluation of models on TRWIKI-67, and BPC for TRNEWS-64. Note that lower is better for both metrics.

We note that PPL needs to be computed with the same count of tokens, otherwise it needs to be normalized in case different tokenization methods are preferred. If not normalized, the metrics would not be comparable (Shoeybi et al., 2019).

	TRWIKI-67		TRNEWS-64	
	#PARAM	PPL	#PARAM	BPC
ADAP.TRANS	92M	14.64	38M	1.024
SHA-RNN	87M	**12.54**	53M	**0.938**

Table 3: Results of language modeling baseline models, with their no of parameters. Perplexity (PPL) is reported for TRWIKI-67, and Bits-per-char (BPC) for TRNEWS-64, on their test sets.

Baselines We consider two baseline models of different families. The first one is Single Headed Attention - RNN (SHA-RNN) (Merity, 2019), which is a Recurrent Neural Network-based language model, and the second is Adaptive Transformer (ADAP.TRANS) (Sukhbaatar et al., 2019), which is based on Transformer architecture (Vaswani et al., 2017). In Table 3, we provide the results of these models, which we train separately on TRWIKI-67 and TRNEWS-64 datasets. It is notable that unlike the case for the English language (Merity, 2019), SHA-RNN performed better than Adaptive Transformer for both of the presented Turkish datasets.

4.2 Machine Translation

Machine translation is the problem of translating a piece of text from one language to another. Over the years, neural machine translation models have become dominant, especially in low resource settings, benefiting from transfer learning (Zoph et al., 2016). In this work, we focus on evaluating neural machine translation models for translation between English and Turkish languages. We provide the results of three different baselines on two datasets.

Datasets The first dataset we evaluate is the Turkish-English subset of WMT16. This dataset was presented at the first Conference of Machine Translation (WMT)[3], it consists of manually translated Turkish-English sentence pairs. The second one is the Turkish-English subset of Multilingual Speech Translation Corpus (MuST-C) (Di Gangi et al., 2019). This corpus was extracted from movies and TV shows subtitles. We present the statistics of both datasets in Table 4.

	#Sentences	#Words
Turkish		
MUST-C	236K / 1.3K / 2K	3.4M / 19K / 33K
WMT-16	205K / 1K / 3K	3.6M / 14K / 44K
English		
MUST-C	236K / 1K/ 2K	4.6M / 26K / 45K
WMT-16	205K / 1K / 3K	4.4M / 19K / 58K

Table 4: Results of machine translation baselines. Each cell represents the (Train / Validation / Test) values of the datasets in the corresponding row. WMT-16 and MUST-C refer to Turkish-English subsets.

Evaluation We evaluate our models on the relevant test sets for translation in both directions. We utilize BLEU Score (Papineni et al., 2002) for the assessment of translation quality.

	WMT-16		MUST-C	
	←	→	←	→
from scratch				
CONVS2S (180M)	13.22	12.78	21.79	13.3
TRANS. (58M)	17.29	15.72	27.01	15.52
pre-trained				
MBART50 (680M)	**24.17**	**18.54**	**32.97**	**19.61**

Table 5: BLEU scores of machine translation baselines. Results are provided for translations in both directions (En↔Tr).

Baselines In this task, we train three different models. First, we train a TRANSFORMER (Vaswani et al., 2017) with the same settings for the encoder and the decoder parts. Where we use 6 layers, with 4 attention heads each, and hidden size of 512. Second, we utilize the Convolutional sequence-to-sequence CONVS2S model (Gehring et al., 2017) following the same settings. The last model is mBART 50 (Tang et al., 2020), a multilingual model pre-trained on 50 different languages, which we fine-tune for each dataset separately.

In Table 5 we present BLEU score of the models on each translation dataset in both directions. The benefit of pre-training can be seen in the case of MBART50, where it outperforms the counterparts that we train from scratch.

[3] http://www.statmt.org/wmt16/

4.3 Named-Entity Recognition (NER)

We include the Named-Entity Recognition (NER) task in our set of benchmarks, as it has an essential role in NLP applications. In this task, words representing named-entities are detected in the text input and assigned one of the predefined named-entity classes such as *Person* or *Location* (Chinchor and Robinson, 1998). We benchmark three different models on two NER datasets for Turkish and compare our work with previous work.

Datasets The first dataset we use is MILLIYET-NER (Tür et al., 2003), which is a set of manually, annotated news articles from the Turkish Milliyet news resource[4]. The second is the Turkish subset of the semi-automatically annotated Cross-lingual NER dataset WIKIANN or (PAN-X) (Pan et al., 2017), which consists of Turkish Wikipedia articles. Both datasets have three entity classes as shown in Table 6.

	Training	Validation	Test
WIKIANN			
Location	9679	5014	4914
Organization	7970	4129	4154
Person	8833	4374	4519
Total words	149786	75930	75731
MILLIYET-NER			
Location	8821	942	1126
Organization	8316	842	873
Person	13290	1400	1603
Total words	419996	45532	49595

Table 6: Distribution of Named entities over classes in MILLIYET-NER and WIKIANN datasets.

Evaluation NER systems are evaluated based on their capability of finding named entities with their correct boundaries and classes. Following previous work on Turkish NER (Yeniterzi, 2011; Şeker and Eryiğit, 2012), we report the CoNLL F-measure metric (Tjong Kim Sang, 2002) to assess our NER baselines.

Baselines We train three different baseline models for this task. One with no pre-trained embeddings, which utilizes bi-directional Long Short Term Memory with Conditional Random Fields (BILSTM-CRF) (Panchendrarajan and Amaresan, 2018). The remaining two models employ pre-trained representations from BERT (Devlin et al., 2019). In one of the models, we investigate the benefit of adding a CRF layer on top of BERT. As for the pre-trained BERT model, we use BERTURK base, which is pre-trained on a large Turkish corpus (Schweter, 2020).

	MILLIYET	WIKIANN
(Yeniterzi, 2011)	91.56	-
(Şeker and Eryiğit, 2012)	91.94	-
(Güngör et al., 2018)	93.37	-
BILSTM-CRF	95.54	**93.8**
BERTURK	95.31	92.82
BERTURK-CRF	**96.48**	93.07

Table 7: Evaluation results (CoNLL F-measure) of NER models on test sets.

In Table 7, we provide the evaluation results (CoNLL F-measure) for the three baselines on both datasets' test sets. Additionally, we compare our results with previous work of (Yeniterzi, 2011; Şeker and Eryiğit, 2012; Güngör et al., 2018) on MILLIYET-NER dataset.

4.4 Sentence Segmentation

Sentence segmentation is the task of detecting sentence boundaries in a given article. Despite its fundamental place in the NLP pipelines, sentence segmentation attracts little interest. Common approaches are rule-based systems that rely on cues such as punctuation marks and capital letters (Jurafsky and Martin, 2018).

Datasets We present TRSEG-41, a new sentence segmentation dataset for Turkish. This dataset consists of 300 scientific abstracts from (Özturk et al., 2014), 300 curated news articles from TRNEWS-64, and a set of 10K tweets. For the scientific abstracts, our sampling rationale is to maximize the number of abbreviations that reduce the accuracy of the rule-based approaches. As for the news set, we maximize the length of documents and the number of proper name. In the Twitter dataset, we balance the number of multi/single sentence tweets, and preprocess the tweets by replacing all URLs with `http://some.url`, and all user mentions with `@user`.

We manually annotate the sentence boundaries of these articles and present two dataset splits, one for training and development and one for testing and benchmarking. The statistics of the splits can be found in Table 8.

Applying sentence segmentation to user-generated content such as social media posts or

[4] https://www.milliyet.com.tr/

	#Articles	#Sentences	#Words
News	300	6K	102K
Tweets	10K	28K	242K
Abstracts	300	6K	112K
Total	10.6K	40K	456K

Table 8: Statistics of TRSEG-41 dataset.

comments can be quite challenging. To simulate such difficult cases and expose the weaknesses of rule-based methods, we create another version of **TRSEG-41** where we artificially corrupt the boundaries of sentences. This is done by randomly converting them to lowercase or uppercase with 50% probability, or by removing all punctuation marks with 50% probability.

Evaluation Our evaluation procedure is based on the metrics F1 score, Precision, Recall for each segment. Unlike (Wicks and Post, 2021), we evaluate our models with the entire test set, without removing sentences with ambiguous boundaries. Furthermore, in order to highlight the gap in performance, we cross-evaluate our systems on the original and corrupted set.

	F1-score	Precision	Recall
spaCy	0.74 / 0.37	0.76 / 0.48	0.72 / 0.30
Training(Original)			
Ersatz	**0.89 / 0.40**	**0.98** / 0.51	0.81 / **0.33**
Punkt	0.87 / 0.39	0.88 / **0.52**	**0.86** / 0.32
Training(Corrupted)			
Ersatz	**0.88 / 0.40**	**0.97 / 0.51**	0.81 / **0.33**
Punkt	0.85 / 0.39	0.86 / 0.50	**0.84** / 0.31

Table 9: Results of sentence segmentation baselines. Metrics are reported for both corrupted and clean versions of the test set in the Original / Corrupted format.

Baselines For this task, we employ three methods as baseline models. Ersatz, a context-based approach that relies on supervised training (Wicks and Post, 2021), the unsupervised Punkt tokenizer (Kiss and Strunk, 2006), and spaCy Sentencizer tool (Montani et al., 2021). While Ersatz utilizes Transformer (Vaswani et al., 2017) architecture, spaCy Sentencizer is a rule-based sentence boundary detector, whereas Punkt Tokenizer relies on an unsupervised training approach.

We experiment with these models on four different training and testing set combinations, where we train using the original and corrupted training sets separately and test on both test sets. Results are presented in Table 9.In all settings, spaCy Sentencizer is outperformed by its trained counterparts. Among the baselines, Ersatz performed the best. Our experiments show that deep learning models are more robust to corruption in the data.

4.5 Spellchecking and Correction

Spellcheckers are among the most widely used NLP tools. The basic task is to check for misspellings in an input and suggest a set of corrections. Different methods can be employed for error correction, such as looking up words that minimize the edit distance from a dictionary or utilizing probabilistic models with N-grams to suggest the most likely correct word based on the context (Jurafsky and Martin, 2018). In this work, we focus on contextless (single word) spellchecking and correction. We present a new benchmarking dataset for this type of spellcheckers and an efficient dictionary for Turkish.

Datasets We present **TRSPELL-10**, a dataset of 10K words, for benchmarking spellchecking and correction. The dataset consists of tuples of input and correct (gold) words.

To create this dataset, we randomly sample 8500 Turkish words from the TS Corpus Word List (Sezer, 2013, 2017). We create artificial misspellings by applying random insertions, deletions, and substitutions on 65% of the words, where we apply at most two operations on the same word. The remaining 35% of the words are unchanged. Moreover, we add 1K random foreign words, and 500 randomly generated word-like character sequences.

As a quality check of these artificial misspellings, given a list of corrupted words, we ask our annotators to provide us a list of suggestions up to 10 suggestions per word. Their suggestion lists had the gold output 91% of the time.

Evaluation We evaluate spellcheckers' ability to detect misspellings using the macro-averaged F1-Score metric. Additionally, we evaluate their spell correction accuracy (SCA) based on the suggestions provided for misspelled words.

Baselines We take advantage of the agglutinative nature of the Turkish language by developing a Hunspell-based (Trón et al., 2005) dictionary for Turkish. Using a list of 4M words we filter from

	SCA	F1
Hunspell-tr (Zafer, 2017)	48.31	97.45
Zemberek (Akın and Akın, 2007)	**62.69**	93.99
Our Hunspell	55.64	**98.16**

Table 10: Spell correction accuracy (SCA) and macro-averaged F1 scores of spellchecking methods on TRSPELL-10.

Web crawls and Turkish corpora, we optimize the splits that minimize the size of the root dictionary and the affix list.

We compare this dictionary to hunspell-tr (Zafer, 2017), another Hunspell-based Turkish dictionary, and to Zemberek spellchecker (Akın and Akın, 2007), which is designed based on morphological features of the Turkish language. As shown in Table 10, our dictionary surpasses other baselines in terms of error detection. However, Zemberek's correction accuracy is higher compared to the Hunspell based methods.

4.6 Summarization

Abstractive text summarization is the task of generating a short description (summary) of an article (longer text). Formally, given a sequence of tokens (input article) $X = (x_1, x_2, ..., x_n)$ and its summary $Y = (y_1, y_2, ..., y_m)$, the main task is to model the conditional probability: $P(Y|X) = \prod_{i=1}^{m} P(y_i|y_{<i}, X)$.

For this task, we work on the Multi-lingual Summarization (Mlsum) dataset (Scialom et al., 2020) and present state-of-the-art summarization results for Turkish.

Datasets Mlsum is a multi-lingual dataset for abstractive summarization. This dataset consists of a large set of crawled news articles with their abstracts in multiple languages. We focus on the Turkish subset of Mlsum.

	Original	Cleaned
Avg. article length	259.1	258.4
Avg. summary length	18.5	18.3
Splits		
Training	249277	246490
Validation	11565	10852
Test	12775	11897
Total	273617	269239

Table 11: Statistics of the Turkish subset of Mlsum.

We found 4378 duplicated instances and 12 overlapping instances among the splits while assessing the dataset's quality. We remove these instances from the dataset for a more accurate evaluation and evaluate our models on both the original and the cleaned sets. In Table 11, we provide some statistics about both sets, before and after the deduplication.

Evaluation To assess the quality of the generated summaries, we use the N-gram co-occurrence-based Rouge-L (Lin, 2004) and Meteor (Banerjee and Lavie, 2005) metrics. We report two different results for each model, one on the original test set and another on the cleaned set.

	Rouge-L	Meteor
(Scialom et al., 2020)	32.90/ –	26.30/ –
TrBart (120M)	35.54/35.08	26.47/25.81
mBart50 (680M)	39.21/38.47	30.84/30.36
mT5-Base (220M)	**39.92/38.76**	**31.72/31.47**

Table 12: Evaluation of different models on Mlsum test set along with their no of parameters. Metrics are calculated for both (Original/Cleaned) test sets.

Baselines As a baseline model for summarization, we present TrBart, a Seq2Seq Transformer (Vaswani et al., 2017) trained following the configuration of BART Base (Lewis et al., 2020), which is a state-of-the-art model for abstractive summarization in English.

Moreover, we fine-tune two different pre-trained models. The first model is Multilingual BART (mBart50) (Tang et al., 2020), which is pre-trained on data from 50 different languages. The second model is Multilingual Text to Text Transformer (mT5-Base) (Xue et al., 2021). As shown in Table 12, all models performs better than the best proposed baseline model (Scialom et al., 2020), which follows UniLM architecture (Dong et al., 2019).

4.7 Text Classification

Text classification can be utilized in several applications such sentiment analysis or topic identification. In this task we take a sequence of text as an input, and output a probability distribution over arbitrary number of classes. In our work on Turkish we benchmark three models on two datasets from different domains.

Datasets We work on the news categorization (News-Cat) dataset (Amasyalı and Yıldırım, 2004). In this dataset, news articles are labeled with one of the following five categories *health, sports, economy, politics, magazine*. There is no splits provided in the original work for New-Cat dataset. Hence we construct our own splits.

	OffensEval	News-Cat
Avg. article length	8.5	227.3
#Classes	2	5
Splits		
Training	28000	750
Validation	32777	150
Test	3515	250
Total	64292	1150

Table 13: Statistics of News-Cat and OffensEval dataset splits.

The second dataset is the corpus of Offensive Speech Identification in Social media (OffensEval) (Çöltekin, 2020). This dataset was collected from Twitter, where the tweets are annotated for offensive speech with *offensive*, or *non-offensive* labels. We choose these datasets for benchmarking since they vary in domain, average article length.

Evaluation We evaluate our baseline models using the F1 score. We use the macro averaged variant to account for the imbalance in classes within the datasets.

	OffensEval	News-Cat	Avg.
BiLSTM	0.747	0.808	0.777
CNN-Text	0.751	0.883	0.817
BERTurk	**0.823**	**0.944**	**0.883**

Table 14: Evaluation results (macro averaged F1-Score) of our baseline models for text classification task. The last column represent the average F1-scores of each model.

Baselines We measure the performance of three deep learning models—one with pre-training and two with none. The pre-trained model is the Bert (Devlin et al., 2019) based Turkish pre-trained (BERTurk) model (Schweter, 2020). The remaining two models employ randomly initialized embeddings of size 256. In one of them we use two layers of Bidirectional LSTM (BiLSTM) (Hochreiter and Schmidhuber, 1997) with a hidden size of 256. In the other model (CNN-Text), we use Convolutional Neural Networks for Sentence Classification (Kim, 2014) with 32 filters instead of 2.

Looking at F1 scores of the models In Table 14, we can observe the clear advantage of the pre-trained BERTurk model over BiLSTM and CNN-Text.

5 Conclusion

We believe that while some human languages such as Turkish do not fall under the definition of under-resourced languages, they are shown little interest as a result of the lack of organized benchmarks and baselines. To address this problem, we presented Mukayese, a comprehensive set of benchmarks along with corresponding baselines for seven different tasks: Language Modeling, Machine Translation, Named Entity Recognition, Sentence Segmentation, Spell Checking and Correction, Summarization, and Text Classification, as well as four new benchmarking datasets in Turkish for Language Modeling, Sentence Segmentation, and Spell Checking and Correction. For future work, the same methodology can be followed to include more NLP tasks such as Dependency Parsing, Morphological Analysis and other tasks.

We hope that Mukayese sets an example and leads to an increase in efforts on under-researched languages.

References

Ahmet Afsin Akın and Mehmet Dündar Akın. 2007. Zemberek, an open source nlp framework for turkic languages. 10:1–5.

MF Amasyalı and T Yıldırım. 2004. Otomatik haber metinleri sınıflandırma. In *2004 12th Signal Processing and Communications Applications Conference (SIU)*, pages 224–226. IEEE.

Giusepppe Attardi. 2015. Wikiextractor. `https://github.com/attardi/wikiextractor`.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

N. Chinchor and P. Robinson. 1998. Appendix E: MUC-7 named entity task definition (version 3.5). In *Seventh Message Understanding Conference (MUC-7):*

Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998.

Çağrı Çöltekin. 2020. A corpus of Turkish offensive language on social media. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6174–6184, Marseille, France. European Language Resources Association.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mattia A Di Gangi, Matteo Negri, and Marco Turchi. 2019. One-to-many multilingual end-to-end speech translation. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 585–592. IEEE.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1243–1252. JMLR.org.

Onur Güngör, Suzan Üsküdarlı, and Tunga Güngör. 2018. Recurrent neural networks for turkish named entity recognition. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Marcus Hutter. 2006. The human knowledge compression prize.

Chip Huyen. 2019. Evaluation metrics for language modeling. *The Gradient*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Daniel Jurafsky and James H Martin. 2018. Speech and language processing (draft). *https://web. stanford. edu/˜ jurafsky/slp3*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Fernando Martínez-Plumed, Pablo Barredo, Seán Ó hÉigeartaigh, and José Hernández-Orallo. 2021. Research community dynamics behind popular ai benchmarks. *Nature Machine Intelligence*, 3(7):581–589.

Stephen Merity. 2019. Single headed attention rnn: Stop thinking with your head. *Computing Research Repository*, arXiv:1911.11423. Version 2.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Ines Montani, Matthew Honnibal, Matthew Honnibal, Sofie Van Landeghem, Adriane Boyd, Henning Peters, Paul O'Leary McCann, Maxim Samsonov, Jim Geovedi, Jim O'Regan, György Orosz, Duygu Altinok, Søren Lind Kristiansen, Roman, Explosion Bot, Leander Fiedler, Grégory Howard, Wannaphong Phatthiyaphaibun, Yohei Tamura, Sam Bozek, murat, Mark Amery, Björn Böing, Pradeep Kumar Tippa, Leif Uwe Vogelsang, Bram Vanroy, Ramanan Balakrishnan, Vadim Mazaev, and GregDubbin. 2021. explosion/spaCy: v3.2.0: Registered scoring functions, Doc input, floret vectors and more.

Seçil Özturk, Bülent Sankur, Tunga Gungör, Mustafa Berkay Yilmaz, Bilge Köroğlu, Onur Ağin, Mustafa İşbilen, Çağdaş Ulaş, and Mehmet Ahat. 2014. Turkish labeled text corpus. In *2014*

22nd Signal Processing and Communications Applications Conference (SIU), pages 1395–1398. IEEE.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Rrubaa Panchendrarajan and Aravindh Amaresan. 2018. Bidirectional LSTM-CRF for named entity recognition. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stefan Schweter. 2020. Berturk - bert models for turkish.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

Gökhan Akın Şeker and Gülşen Eryiğit. 2012. Initial explorations on using CRFs for Turkish named entity recognition. In *Proceedings of COLING 2012*, pages 2459–2474, Mumbai, India. The COLING 2012 Organizing Committee.

Taner Sezer. 2013. Ts corpus: Herkes İçin türkçe derlem. *Proceedings 27th National Linguistics Conference*, pages 217–225.

Taner Sezer. 2017. Ts corpus project: An online turkish dictionary and ts diy corpus. *European Journal of Language and Literature*, 9(1):18–24.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *Computing Research Repository*, arXiv:1909.08053. Version 4.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 331–335, Florence, Italy. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *Computing Research Repository*, arXiv:2008.00401.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Viktor Trón, Gyögy Gyepesi, Péter Halácsky, András Kornai, László Németh, and Dániel Varga. 2005. Hunmorph: Open source word analysis. In *Proceedings of Workshop on Software*, pages 77–85, Ann Arbor, Michigan. Association for Computational Linguistics.

Gökhan Tür, Dilek Hakkani-TüR, and Kemal Oflazer. 2003. A statistical information extraction system for turkish. *Natural Language Engineering*, 9(2):181–210.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, page 6000–6010.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, page 3266–3280. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Rachel Wicks and Matt Post. 2021. A unified approach to sentence segmentation of punctuated text in many languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3995–4007, Online. Association for Computational Linguistics.

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.

Reyyan Yeniterzi. 2011. Exploiting morphology in Turkish named entity recognition system. In *Proceedings of the ACL 2011 Student Session*, pages 105–110, Portland, OR, USA. Association for Computational Linguistics.

Harun Reşit Zafer. 2017. hunspell-tr. `https://github.com/hrzafer/hunspell-tr`.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.