#Turki$hTweets: A Benchmark Dataset for Turkish Text Correction

Asiye Tuba Koksal[1], **Ozge Bozal**[1,2], **Emre Yurekli**[1], **Gizem Gezici**[1,3]
[1]Huawei R&D Center, Istanbul, Turkey
[2]Bogazici University, Istanbul, Turkey
[3]Sabanci University, Istanbul, Turkey
{asiye.tuba.koksal, ozge.bozal, emre.yurekli, gizem.gezici}@huawei.com

Abstract

#Turki$hTweets is a benchmark dataset for the task of correcting the user misspellings, with the purpose of introducing the first public Turkish dataset in this area. #Turki$hTweets provides correct/incorrect word annotations with a detailed misspelling category formulation based on the real user data. We evaluated four state-of-the-art approaches on our dataset to present a preliminary analysis for the sake of reproducibility. The annotated dataset is publicly available at https://github.com/atubakoksal/annotated_tweets.

1 Introduction

The extensive use of social media platforms such as Twitter, Facebook, forums, and blogs has created a new form of written language, which is full of intentional and unintentional misspellings as well as newly-emerged slang words and abbreviations. This new type of language poses significant challenges for various natural language processing (NLP) tasks, mostly requiring properly written textual content for analysis. Therefore, text normalization, i.e., transforming non-standard words into their standard forms and spelling correction, i.e., correcting unintentional spelling errors, have become indispensable pre-processing stages. The pre-processing phase is known to boost the model performance for the various NLP applications, including but not limited to POS tagging, sentiment classification and search[1].

Although correcting misspelling errors is crucial for NLP applications, it is generally not straightforward and even challenging for morphologically rich languages. There exist many different surface forms of a single word in highly agglutinative languages. Specifically for Turkish, suffixes should also comply with the vowels and the last letter of the word. This leads to many different variations of a single word, thereby increasing the possibility of misspelling errors. Previous studies adapted for English do not fit and thus, there is the need for resources tailored particularly for these languages. Currently, there is no publicly available dataset in this area that can be used for model evaluation with reproducible results.

In the online platforms, there exist not only intentionally misspelled words but also unintentional spelling errors, both constituting out-of-vocabulary (OOV) words in the textual content. Intentional user misspellings are quite frequent, particularly in online media and these errors highly vary depending on the intention of use. Therefore, a more fine-grained analysis of the error types is required by categorizing the errors and further including them in the evaluation dataset for a proper model evaluation. In this way, a particular model can be assessed whether it provides generalizable results for the corresponding language.

Based on these, we introduce a new Turkish dataset by categorizing, annotating and correcting the distinct misspelling types in text. Moreover, we make a fine-grained evaluation of the selected state-of-the-art approaches in literature for reproducibility purposes. To the best of our knowledge, this is the first attempt that introduces a public dataset including the detailed misspelling category formulation with the purpose of providing a reproducible evaluation results on the existing approaches. We hope that the fine-grained analysis of selected models in this work serves as an exemplary usage of the dataset. Most similar work to ours is Eryiğit and Torunoğlu-Selamet (2017) in which authors define rules for correcting the misspelling errors present in social media content. The authors compare the proposed model with the selected state-of-the-art approaches on their dataset which is not

[1]Query correction can help to increase the search performance by correctly understanding user intent.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4190–4198

publicly available to researchers. Hence, we created a benchmark dataset by randomly selecting and annotating Turkish tweets from a public dataset of 20M tweets[2].

Our contributions in this work are as follows.

- We provide a real dataset such that misspellings are created by real users,
- We propose a systematic formulation for error categorization of OOV words in a real dataset,
- We show a fair evaluation of the selected models on the same benchmark dataset, for the sake of reproducibility.

The paper is structured as follows. In Section 2, we give related work. In Section 3, we provide the details about data preparation & analysis. In Section 4, we show the evaluation results. Finally, in Section 5 we conclude the paper.

2 Related work

Research studies in spelling correction of Turkish text date back to 1990s. However, each work has carried out its own evaluation data generation process and none of these datasets are publicly available. In early spelling correction studies, synthetic datasets were used for evaluation. Oflazer (1996) used words collected from Turkish corpus that are perturbed such that the words and their correct forms are 1, 2 or 3 edit distances apart. Büyük et al. (2019), Gupta (2020) and Büyük (2020) created a synthetic dataset composed of misspelled words with 1 edit distance. Büyük (2020) also used a non-public dataset manually annotated by Torunoglu-Selamet et al. (2016) for a better comparison. There are also other works that used real datasets. Oflazer and Güzey (1994) evaluated their model on incorrect words in Turkish text which are mostly 1 edit distance apart. Torunoglu-Selamet et al. (2016) manually annotated words from social media text excluding the intentional mistakes such as words without vowels; they separated the task of text normalization and correction of unintentionally misspelled words. Bölücü and Can (2019) used an open-source morphological analyzer to extract incorrect words from BOUN corpus (Sak et al., 2008) which is composed of newspaper and website textual content.

With the rise of social media, new text style has emerged: micro-blogging. Those text sources have their own jargon including the intentional and unintentional misspellings. Torunoğlu-Selamet and Eryiğit (2014)'s work focused on text normalization in Twitter. They manually aligned 1200 tweets in which some of the tokens are in one-to-many alignment. Researchers used this dataset for evaluation of their proposed rule-based model. Çolakoğlu et al. (2019) used the same dataset, except this, they manually annotated another Twitter dataset for model evaluation. Nonetheless, the dataset introduced in Torunoğlu-Selamet and Eryiğit (2014) is not open to the research community, it can only be obtained upon request under some restricted license constraints.

In this work, we propose a new benchmark dataset composing of real Turkish tweets with misspelling annotations for different types of OOV words.

3 Data preparation

We used a public dataset of 20M Turkish tweets written by real users to create the benchmark dataset. First, we applied some pre-processing steps such as cleaning up the lines with meta-information like timestamps, URLs, usernames, etc. to provide one sentence per line format. After that, we had 23M sentences. Then, out of these 23M sentences, 2000 sentences, including at least one out-of-vocabulary word were randomly selected. To check if there is an OOV word in a sentence, we used TRMorph (Çağrı Çöltekin, 2010), an open-source Turkish morphological analyzer. We first tokenized every sentence on each line, using the TRmorph's tokenizer and sent each token to the TRmorph for morphological analysis. If the TRMorph achieved to provide an analysis for a given token, then the token was assumed to be a correct Turkish word, i.e., in-vocabulary (IV) word for the rest of the paper, otherwise incorrect as referred to as OOV. In this way, we guaranteed at least one OOV word in each sentence obeying one-to-one token alignment. The data statistics and all the details about the error annotation & correction process are provided in Section 3.1, 3.2 and 3.3.

3.1 Preprocessing

We first filtered the appropriate tweets for the annotation process. We have three main criteria for the appropriateness of a given tweet: i. being written in Turkish, ii. forming a full sentence, iii. including at least one misspelled word. There are

[2]https://www.kemik.yildiz.edu.tr/data/File/20milyontweet.rar

many homonymic words in Turkish, the meaning of which can only be inferred when used in a full sentence. Similarly, some orthographic errors such as unintentional character mistakes can only be solved in a context, due to word-sense disambiguation problem. Therefore, we only accepted full-sentence tweets in our dataset. Also, we removed the tweets which contain only hashtags or emojis from our analysis (no correction is necessary).

The use of non-canonical forms of punctuations, e.g., emojis, repetition of punctuations, is quite common in tweets, which is in fact not an orthographical error. All kinds of punctuation and emojis were replaced with white-space in selected tweets, except for these: i. the apostrophe, since it is used to separate some suffixes from proper nouns and deleting it would be an orthographic error and ii. the number sign (#), since this sign indicates hashtags in tweets; it is necessary to keep it to differentiate any word from a hashtag word. We left the numbers as they appeared and annotated them with IV tag, unless there is a misspelling caused by suffixes added to the numbers. All words were converted to lowercase (including formal abbreviations, foreign words and initial words of the sentences), except for the correctly spelled proper nouns. In a word showing enthusiasm, repetitive characters were also left unchanged and considered to be intentional character mistakes. There were misspellings in the dataset regarding compound words, such that some words should have been typed separately, while others adjacently. We added "|" character to indicate a white-space for the token alignment where such errors occurred. For sample instances, please refer to "Separation Error" and "Adjacent Error" in Table 6 in Appendix A.2.

3.2 Data annotation & correction

For the categorization of OOV words, we have been inspired by Aw et al. (2006); Han et al. (2012), which proposed well-defined distinction of English OOV words in terms of whether they need any normalization. We also consulted Han and Baldwin (2011); Beaufort et al. (2010); Pamay et al. (2015); Eryiğit and Torunoğlu-Selamet (2017) in grouping the error types of OOV words.

The annotation of the dataset was completed after examining the different error types present in Turkish tweets. Then, we referred to the authorized dictionary and Turkish spelling rules dictated by the Turkish Language Institution (TDK) for the data correction. Three annotators fulfilled the annotation and correction process accordingly, then the final decisions on the error types were made by consensus.

The error types used for annotation are all mutually exclusive and fully cover all kinds of errors in the dataset, i.e., no additional error type can be found to a misspelling in words of a Turkish tweet. There were both syntactic and semantic errors. We determined thirteen different subgroups considering orthographic spelling errors, intentional errors, non-lexical words derived for social media jargon and slang words. Detailed explanations for each error type can be found in Appendix A.1.

The tokens were tagged with IV or OOV based on the TDK Turkish Dictionary[3]. If a token was tagged with OOV, then one of the error types shown in Table 1 was assigned to this token as well. Furthermore, if a correction was necessary for the token, then it was also assigned an additional tag of *ill_formed*, otherwise *well_formed*.

Tokens were allowed to have multiple tags and the data statistics given in Table 1 are based on the occurrences of the *individual* tags in the dataset. Several examples from the dataset corresponding each error tag can be found in Table 6 Appendix A.2.

3.3 Data statistics

The dataset consists of 2000 sentences and 16878 tokens. Each token has corresponding error tags, where the tokens and tags are aligned with each other. There exist 9713 unique tokens and 6488 of them are OOV tokens. The percentages of different error types in the OOV tokens are given in Table 1. The most frequent error type is the deasciification, while the least frequent one is the phonetic substitution. Since the dataset consists of real user tweets, it also gives us some hints about users' general misspelling tendencies in Turkish social media.

Among 2000 sentences, 77% of them have more than one error and 59% of all sentences contain multiple error types.

4 Experiments

The performance of a text correction model can be evaluated with the following metrics, *correction rate over the misspelled words* and *non-corruption rate over already correct words*. For this reason, we built a two-step pipeline for the text correction:

[3]https://sozluk.gov.tr/

Error Types		P.(%)
Deasciification	*ill-formed*	44.94
Accent	*ill-formed*	11.22
Proper Name	*ill-formed*	9.20
Intentional Char	*ill-formed*	9.02
Seperation	*ill-formed*	7.68
Foreign Word	well-formed	4.92
Unintentional Char	*ill-formed*	4.69
Social Media Phrase	well-formed	2.50
Abbreviation	well-formed	2.37
Adjacent	*ill-formed*	1.36
Neologism	well-formed	0.96
Vowel	*ill-formed*	0.63
Phonetic Substitution	*ill-formed*	0.52

Table 1: The percentage distribution of error types over OOV words in the dataset.

i. OOV word detection and ii. word correction. In the first part, we aimed to detect the OOV words for the correction step, thus preventing unnecessary modifications in IV words. For this purpose, we compared the performance of two morphological analyzers on finding the OOV words in the dataset. As the second step, we compared the correction and non-corruption rates of several text correction models on the OOV words detected by the better performing analyzer from the previous step. In the experiments, we used TRMorph's morphological analyzer, an open-source Turkish NLP library Zemberek[4] and our implementations for the rest of the models. These experiments were conducted on 9223 unique words which consist of *ill-formed* OOV tokens (refer Table 1) and IV tokens from the dataset.

4.1 Morphological Analysis and OOV Detection

We compared TRmorph's and Zemberek's morphological analyzers in terms of two aspects: 1. What percentage of the words that are considered to be OOV are true OOV words, and 2. What percentage of the true OOV words were identified. The corresponding precision and recall values and F1 scores can be found in Table 2 (For the same analysis of the IV words, please see Table 5 in Appendix A.2).

[4]https://github.com/ahmetaa/zemberek-nlp

	Precision	Recall	F1-score
TRmorph	0.977	0.822	**0.893**
Zemberek	0.985	0.748	0.850

Table 2: *Out-of-vocabulary* word detection results of the morphological analyzers.

4.2 Correction of OOVs

In this section, we will briefly mention the models used in the experiments. For the preliminary results, we selected the frequently used models in this area, the source codes of which are publicly available, except the model described in Section 4.2.3. In Table 3, the models were evaluated on the OOV words dataset detected by TRmorph as described in Section 4.1, since it's F1 score is better than Zemberek (see Table 2). Note that the resulting OOV word dataset is noisy in the sense that, it contains some IV words which were misidentified as OOV by TRMorph.

4.2.1 Zemberek

In Table 3, we evaluated Zemberek's normalization module composing of spell checker (first model) and noisy text normalizer (second model). The spell checker module suggests multiple words for a given OOV word. In this experiment, the highest-ranked suggestion was accepted as the correction of the given OOV word. Before testing these models, each token was checked, whether it had repetitive characters. If a character was consecutively repeating more than twice, it was normalized to one character of itself.

4.2.2 Edit Distance

We implemented the edit distance algorithm which returns the most probable candidate word in maximum 2 edit distance. The model uses the METU Turkish Corpus (Say et al., 2002) to retrieve the possible candidates and chooses the word with minimum edit distance and the highest frequency information in the corpus.

4.2.3 Rule-based Pipeline

In this part, we implemented a rule-based model, which is similar to the cascaded model proposed in Torunoğlu-Selamet and Eryiğit (2014). We note that our model is not the exact replication of the cascaded model[5].

The steps are defined as follows.

[5]The source code is not publicly available.

1. Check if a given word is OOV or IV with the TRmorph's morphological analyzer. If it is IV, then the process returns the word and terminates, otherwise goes to the next step.
2. Remove recurrent characters (e.g. geeeeeel → gel), then try step 1.
3. Deasciify the token (e.g. canim → canım), try step 1.
4. Deaccent the token (e.g. gelmiyom → gelmiyorum), then try step 1.
5. Suggest a possible corrected form of the word using a look-up table.

For steps 2 and 3, we used regular expressions to correct the misspelled words, as illustrated in the steps above. For step 3, we used an open-source deasciifier module[6], which translates the ASCII characters into their Turkish counterparts (e.g., o → ö, s → ş). For the final step, we prepared a look-up table consisting of 1.9M IV tokens in METU (Say et al., 2002) and BOUN (Sak et al., 2008) corpora using TRmorph's analyzer. We removed all vowels from each token (if a token begins with a vowel, we kept it) and created consonant skeleton & possible full form pairs. One sample entry of the look-up table for the word *glyrm*: *geliyorum*, *gülüyorum*. Each misspelled word was searched in this look-up table to retrieve its consonant skeleton & possible full forms pair. Then, using the vowels and their sequence in the misspelled word, the word's possible correct form was returned from its possible full forms.

4.3 Discussion

A successful text corrector model is considered to have a high correction rate on the misspelled words (OOV words) and a high non-corruption rate on the already correct words (IV words). The dataset we used for the comparative evaluation of the selected models is noisy as explained in Section 4.2. Therefore, we compared the models in terms of their correction rate on actual OOV words (C in Table 3), non-corruption rate on IV words (NC in Table 3), and the overall accuracy (Acc in Table 3) calculated on all of the tokens (IV and OOV). To better identify the reasons behind the differences in performance results, we made a detailed analysis of each model in Table 4 on each *ill-formed* error type listed in Table 1.

[6]https://github.com/emres/turkish-deasciifier

Model	C	NC	Acc
Zemberek-Spellchecker (ZS)	0.409	**0.741**	0.415
Zemberek-Normalizer (ZN)	**0.714**	0.638	**0.713**
Edit distance (ED)	0.373	0.476	0.375
Rule-based (RB)	0.602	0.724	0.605

Table 3: Correction (C), non-corruption (NC) and accuracy (Acc) rates of several models on words which are identified as OOV by TRmorph in section. 4.1.

	ZS	ZN	ED	RB
Accent	0.295	**0.608**	0.226	0.399
Adjacent	0.0	0.143	**0.531**	0.0
Deascii	0.407	**0.871**	0.433	0.858
Int. char.	0.667	**0.683**	0.448	0.361
Phon. sub.	**0.435**	0.391	0.391	0.0
Proper	**0.406**	0.009	0.0	0.004
Separation	0.0	**0.479**	0.0	0.0
Unint. char.	**0.534**	0.507	0.507	0.137
Vowel	0.045	**0.636**	0.091	0.182

Table 4: Accuracy results of the models from Table 3 on each *ill-formed* error type.

5 Conclusion

We introduced a benchmark dataset for Turkish text correction by annotating the real users' misspellings in Turkish tweets. We categorized the error types and corrected them accordingly. The dataset can be used in various NLP applications, especially for the social media platforms. The error category formulation can also be used for other tasks like query correction in search, which highly affects the search performance.

We hope that our work will be a valuable contribution to the Turkish research community, being the first work proposing a benchmark dataset with a fine-grained and fair comparative evaluation of some of the state-of-the-art models. As future work, we plan to analyze existing models' deficiencies elaborately and establish new models performing better on our benchmark dataset for distinct error types.

References

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 33–40. Association for Computational Linguistics.

Richard Beaufort, Sophie Roekhaut, Louise-Amélie Cougnon, and Cédrick Fairon. 2010. A hybrid rule/model-based finite-state framework for normalizing sms messages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 770–779. Association for Computational Linguistics.

Osman Büyük. 2020. Context-dependent sequence-to-sequence turkish spelling correction. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(4).

Osman Büyük, Mustafa Erden, and Levent M Arslan. 2019. Context influence on sequence to sequence turkish spelling correction. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE.

N. Bölücü and B. Can. 2019. Context based automatic spelling correction for turkish. In *2019 Scientific Meeting on Electrical-Electronics Biomedical Engineering and Computer Science (EBBT)*, pages 1–4.

Talha Çolakoğlu, Umut Sulubacak, and Ahmet Cüneyd Tantuğ. 2019. Normalizing non-canonical Turkish texts using machine translation approaches. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Florence, Italy. Association for Computational Linguistics.

Gülşen Eryiğit and Dilara and Torunoğlu-Selamet. 2017. Social media text normalization for turkish. *Natural Language Engineering*, 23(6):835–875.

Prabhakar Gupta. 2020. sc. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pages 116–122. IEEE.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 368–378. Association for Computational Linguistics.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 421–432. Association for Computational Linguistics.

Kemal Oflazer. 1996. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics*, 22(1):73–89.

Kemal Oflazer and Cemaleddin Güzey. 1994. Spelling correction in agglutinative languages. In *Proceedings of the fourth conference on Applied natural language processing*, pages 194–195. Association for Computational Linguistics.

Tuğba Pamay, Umut Sulubacak, Dilara Torunoğlu-Selamet, and Gülşen Eryiğit. 2015. The annotation process of the itu web treebank. In *Proceedings of the 9th Linguistic Annotation Workshop*, pages 95–101.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2008. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *International Conference on Natural Language Processing*, pages 417–427. Springer.

Bilge Say, Deniz Zeyrek, Kemal Oflazer, and Umut Özge. 2002. Development of a corpus and a treebank for present-day written turkish. In *Proceedings of the eleventh international conference of Turkish linguistics*, pages 183–192. Eastern Mediterranean University.

Dilara Torunoglu-Selamet, Eren Bekar, Tugay Ilbay, and Gülsen Eryigit. 2016. Exploring spelling correction approaches for turkish. In *Proceedings of the 1st International Conference on Turkic Computational Linguistics at CICLING, Konya*, pages 7–11.

Dilara Torunoğlu-Selamet and Gülşen Eryiğit. 2014. A cascaded approach for social media text normalization of turkish. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 62–70.

Çağrı Çöltekin. 2010. A freely available morphological analyzer for turkish. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

A Appendix

A.1 Error Tags

A.1.1 Ill-formed OOV tags

Ill-formed refers to the misspelled words that have orthographic or cognitive errors calling for correction. Each of the following tags corresponds to a subcategory of ill-formed words.

- **Deasciification errors** consist of the errors corresponding to incorrect substitution of Turkish characters (*ı, ü, ö, ç, ğ, ş*). Both false usage of ascii characters instead of their Turkish deascii counterparts and vice versa are tagged as deasciification error.

  Examples:

  canim → canım (my dear)

  kemık → kemik (bone)

- **Accent errors** In Turkish, most of the words are pronunciated as they are written. However, this rule is violated in everyday spoken language or by some local accents. Accent errors consist of both cognitive and intentional errors due to such pronunciation of Turkish words.

  Examples:

  geliyom → geliyorum (I am coming)

  bi şey → bir şey (a thing)

  de mi → değil mi (isn't it)

- **Proper name errors** occur when proper nouns start with lowercase letters or an apostrophe is needed to separate a suffix from the proper noun but it lacks.

  Examples:

  ayşe → Ayşe (Turkish proper name)

  13ü → 13'ü

  mehmetin → Mehmet'in (Turkish proper name)

- **Intentional character errors** consist of intentionally mistyped words due to the use of fewer or repetitive characters. This type of errors was categorized as intentional since in this category, the words are written deliberately in an erroneous way by the users for the sake of writing easier/faster or emphasizing an emotion.

  Examples:

  senn → senin (yours)

  gelmeeeeeee → gelme (don't come)

- **Separation errors** occur when the words are written without using a space between them where they should be written separately.

  Examples:

  birşey → bir şey (your)

  bende → ben de (me too)

- **Unintentional character errors** consist of the orthographic errors caused by pressing the wrong character's button on the keyboard (characters in the vicinity of the correct character) or the cognitive errors due to not knowing the correct form of the word.

  Examples:

  kslem → kalem (pen)

  direk → direkt (directly)

- **Adjacency errors** occur when the words are written separately where they should be written without using a space between them.

  Examples:

  hiç biri → hiçbiri (none)

  halbu ki → halbuki (whereas)

- **Vowel errors** occur when the words are written by omitting all the vowels for an/a easier/faster writing.

  Examples:

  snn → senin (your)

  cnm → canım (my dear)

- **Phonetic substitution errors** occur when one or more characters in words are replaced with either their non-Turkish (if the pronunciations are similar) or non-alphabetical symbolic (if the shapes of the characters are similar) counterparts.

  Examples:

  Serqan → Serkan (proper Turkish name)

  ewe → eve (to home)

A.1.2 Well-formed OOV tags

The following 4 tags constitute the well-formed category indicating that we did not correct the words belonging to this category, since there is no correct form of these words in Turkish.

- **Foreign word errors** consist of all foreign words (proper, correctly spelled and misspelled) and derived non-Turkish company, brand names etc. without checking if they are typed correctly. Note that all words considered to be in this category are converted to lowercase.

  Examples:

  direction, directon, justin

  digitürk (company name with misspelling)

  turkcell (company name without misspelling)

- **Social media errors** consist of the words that are vocatives, hashtags etc. that are used in social media texts.

  Examples:

  hahahahah, #resist

- **Abbreviations** consist of the words that are both official abbreviations or commonly used abbreviated forms of Turkish words.

  Examples:

  fb - fenerbahçe (famous Turkish sports club)

  dk - dakika (minute)

- **Neologisms** consist of the derived non-lexical words commonly used in social media texts. Usually such words are derived by using an English word and a Turkish derivational suffix.

  Examples:

  tivit-lemek (tweet-ing)

  hack-lemek (hack-ing)

A.2 Additional Tables

	Precision	**Recall**	**F1-Score**
TRmorph	0.881	0.986	**0.930**
Zemberek	0.840	0.991	0.909

Table 5: Precision and recall values of the morphological analyzers for IV words.

Error Types	**Group Name Tag**	**Wrong**	**Corrected**
Abbreviation	OOV-well_formed-abbr	kib	kib (kendine iyi bak) **(take care of yourself)**
Accent	OOV-ill_formed-accent	geliyom, dimi	geliyorum **(I am coming)**, değil mi **(is that so)**
Adjacent	OOV-ill_formed-joint	bir|kaç	birkaç **(a few)**
Deasciification	OOV-ill_formed-deascii	calismak, gıtmek	çalışmak **(to work)**, gitmek **(to go)**
Foreign Word	OOV-well_formed-foreign	Twitter, iPhone	Twitter, iPhone
Intentional Char	OOV-ill_formed-int	canımmm, haydiii	canım **(sweetheart)**, haydi **(come on)**
Neologism	OOV-well_formed-neologism	kardo	kardo
Phonetic Substitution	OOV-ill_formed-phonetic_sub	geli$im	gelişim **(development)**
Proper Name	OOV-ill_formed-proper	ahmetten	Ahmet'ten
Separation	OOV-ill_formed-sep	herşey	her|şey **(everything)**
Social Media Phrase	OOV-well_formed-social	ahahhaha, sdfsdfsdf, yha, #hashtag	ahahhaha, sdfsdfsdf, yha, #hashtag
Unintentional Char	OOV-ill_formed-unint	gerel, haayt	gerek **(need)**, hayat **(life)**
Vowel	OOV-ill_formed-vowel	tmm, fln	tamam **(okey)**, falan **(so and so)**

Table 6: All error types and name tags of OOV words in the sentence dataset along with the examples and corresponding normalized words.