Towards a Better Understanding of Noise in Natural Language Processing

Khetam Al Sharou*, Zhenhao Li*, Lucia Specia
Language and Multimodal AI (LAMA) Lab, Imperial College London, UK
{k.al-sharou, zhenhao.li18, l.specia}@imperial.ac.uk

Abstract

In this paper, we propose a definition and taxonomy of various types of non-standard textual content – generally referred to as "noise" – in Natural Language Processing (NLP). While data pre-processing is undoubtedly important in NLP, especially when dealing with user-generated content, a broader understanding of different sources of noise and how to deal with them is an aspect that has been largely neglected. We provide a comprehensive list of potential sources of noise, categorise and describe them, and show the impact of a subset of standard pre-processing strategies on different tasks. Our main goal is to raise awareness of non-standard content – which should not always be considered as "noise" – and of the need for careful, task-dependent pre-processing. This is an alternative to blanket, all-encompassing solutions generally applied by researchers through "standard" pre-processing pipelines. The intention is for this categorisation to serve as a point of reference to support NLP researchers in devising strategies to clean, normalise or embrace non-standard content.

1 Introduction

In Natural Language Processing (NLP), "noise" as a concept is not well understood and often described as hard to define (Taghipour et al., 2011). In this paper, we attempt to outline what different types of noise mean in order to support NLP researchers in devising strategies to clean, normalise or embrace unexpected content at either training or inference time.

The term "noise" has been generally used to cover both harmful and meaningful types of non-standard content. We however believe that a distinction should be made between content that should be removed/normalised because it harms NLP systems – henceforth **harmful noise** – and content that should be kept to improve the performance of such systems – henceforth **useful noise**[1]. To clarify further, the usual approach to handling "noise" is to attempt to clean or normalise the data as much as possible, but this content can be as important as any other elements in the data because it carries meaning or intentions. In text classification applications, for example, some types of noise should be kept and taken into account, *e.g.* certain punctuation patterns in sentiment analysis. Furthermore, in generation applications, noise often needs to be transferred to the output, such as emojis in machine translation to preserve sentiment. Moreover, in tasks such as correction of second language learners' essays, errors such as lack of cohesion and incorrect punctuation need to be retained.

Handling noise in NLP has been attracting significant attention especially with the widespread availability of data from social media platforms such as Twitter, Reddit, among others, that created not only new opportunities but also new and different needs (Karpukhin et al., 2019). Text found on these platforms, as well as other user-generated materials, is full of non-standard content (Eisenstein, 2013) which causes problems to NLP systems – typically trained on clean data – and these fail to handle them correctly (Baldwin et al., 2015; Belinkov and Bisk, 2018; Heigold et al., 2018). Dealing with non-standard data has been targeted as a research direction in some areas, for example, machine translation (MT), with methods that aim to be robust to unexpected noise (Belinkov and Bisk, 2018; Karpukhin et al., 2019; Vaibhav et al., 2019). However, work has been mainly limited to a few sources of artificially injected noise types.

[*] Equal contribution

[1] We will keep using the term "noise" to adhere to a term known and used in the existing literature in NLP, but we make a distinction between harmful noise and useful noise.

Proceedings of Recent Advances in Natural Language Processing, pages 53–62
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_007

In this paper, we start by providing an overview of previous work addressing this issue (Section 2); to then propose a definition of noise including cases that needs to be cleaned/normalised (harmful noise) and other cases that needs to be kept (useful noise), and present a comprehensive taxonomy of types of noise (Section 3); and to finally demonstrate the impact of addressing noise in a task-based fashion through experimenting with three tasks, where we compare filtering out harmful noise and keeping useful noise versus cleaning/normalising or keeping all types of noise (Section 4).

Our focus is on linguistic noise rather than other types of noise (such as numeric mislabelling). Therefore, our definitions and categorisation broadly cover (i) text classification/regression tasks, where the noise is in the input text, *e.g.* sentiment analysis; and (ii) text-to-text tasks, where the noise is in the input text and – at training time – can also be in the output text, *e.g.* machine translation.

2 Related Work

Noise in NLP has typically been defined in context of a specific dataset, intermediate or end-task, rather than in general. For example, noise is limited to sentence pairs "not being parallel" in parallel corpus filtering (Taghipour et al., 2011). Han and Baldwin (2011) describe their task as lexical normalisation of "ill-formed words", where the definition is narrowed down to "instances of typos, ad hoc abbreviations, unconventional spellings, phonetic substitutions and other causes of lexical deviation found on Twitter and SMS messages". Subramaniam et al. (2009) define text normalisation where noise becomes "any kind of difference in the surface form of an electronic text from the intended, correct or original text". For an end task where the focus is on improving the performance of a certain task, however, the definition of noise becomes very specific, serving one purpose. For example, noise is referred to as the task of removing unconventional casing by truecasing the data when the aim is to get better vocabulary generalisation. Very often such a blanket, one-size-fits-all pre-processing pipeline is applied without enough consideration to other possible sources of noise, or which sources should actually be kept in the data.

Our work is driven by the lack of studies aimed at defining and providing a comprehensive categorisation of noise types in a way that reflects a better practice and shows sensitivity towards such content. The few exceptions limit their categories to noise found in specific text types. Subramaniam et al. (2009) provide a general overview of noise types in SMS, emails and online chats, while Eisenstein (2013) focuses on identifying types of what is called *bad language* and their possible causes.

Most of the previous studies on handling noise have focused on MT. While some studies have revealed that training with noise increases the robustness of systems towards noisy data, others have indicated that the quality of their systems degraded. This can be attributed to the differences in noise types and their potential impact on the task. For instance, Khayrallah and Koehn (2018) have showed that training MT models on noisy parallel data (such as misaligned sentences or wrong language) leads to weaker systems. However, when models are trained on more specific types of noise such as spelling, profanity, grammar and emoticons, they have been shown to perform better on similar types of noise (Belinkov and Bisk, 2018; Heigold et al., 2018; Ott et al., 2018; Berard et al., 2019; Vaibhav et al., 2019). We note that most of the latter work aims to adapt MT systems built on clean data to noisy test settings by artificially injecting errors to the training set. This is a different problem to the one we discuss in this paper where both training and test sets are taken from the same larger population with naturally occurring noise.

These studies show that different types of noise have different effects in MT and other NLP tasks. However, the types of noise addressed and their effect vary considerably and a consistent definition has not been provided.

Noise is better defined and understood in other related fields. In Automatic Speech Recognition, a distinction is made between what is considered as noise (*e.g.* environmental noise, reverberation) and other speech variations (*e.g.* accent, speaking style and rate, emotional states, speech impairments). Noise is defined as "any unwanted disturbances superposed upon the intended speech signal" (Li et al., 2015), *e.g.* by additive noise (*e.g.* background noise, traffic noise) and convolutional noise (*e.g.* transmission channel distortions, room reverberation, microphone filtering) (Xiong, 2009). In Optical Character Recognition, noise refers to "the error in the pixel value or an unwanted bit pattern, which do not have any significance in the output", where the unwanted bit patterns are introduced by uneven writing surface or poor quality of the data acquisition device and include textured or coloured background (Bansal and Kumar, 2013). Different handwriting however is not seen as noise which needs removing but as a problem which needs to be addressed.

Similar to what has been done in these areas, this paper is an attempt to clearly define and categorise noise in NLP in a way that is not constrained to any particular application, task or dataset. While noise is mainly found in non-standard text, the taxonomy is agnostic to the type of text.

3 A Taxonomy of Noise

We intend to move away from the general labelling of noise as "errors", "difficult data", "ill-formed words", "contamination", "disfluent language" and "corruption" to a more specific one that reflects the current practice which have shifted from denoising to training with noisy text. We therefore look at noise through the same lens as in other related fields where a distinction should be made between unwanted constructs that occur in the text (generally unintentionally), and other (generally intentional) constructs that do not adhere to the conventions and rules of a given language to serve a purpose. Based on this distinction, we define **harmful noise** as any unwanted disturbances that are either harmful or useless or both: Harmful in that it affects the intended meaning of the text and/or the performance of the NLP system; and useless in that their existence does not serve a purpose. **Useful noise**, on the contrary, covers wanted content that serves a purpose and carries meaning that is important for the task and/or for the performance of the NLP system. Our definition does not includes cases where a special language is used in a cryptical fashion, *i.e.* hiding some codes/messages in the text so that only a particular addressee understands the true message. It is also worth noting that it depends on the NLP application under consideration, where harmful noise for one system could be useful noise for another one.

Our taxonomy of noise, displayed in Figure 1, focuses on applications that take sentences as input, as in most NLP tasks. From this work, we exclude types of noise that are related to errors in parallel corpora with text in both input and output such as misalignments, incorrect output, untranslated sentences, among others. The taxonomy is divided into several types and sub-types of noise that are limited to those naturally occurring, produced by humans. Machine-generated errors, *e.g.* stemming from back-translation or intentionally injected, are seen as a way to mimic the naturally occurring noise, and as such not represented separately. The taxonomy is designed to serve as an initial point of reference. It should be seen as language-independent, but it may require modifications to cover very specific language phenomena the authors are not familiar with. Our taxonomy is based on previous works that address different types of noise as well as taxonomies of errors and also on our wide experience improving robustness in NLP. The selection criteria are driven by the intention to include general categories that are applicable to different languages and tasks, and for non-standard content. In what follows, we describe the types and sub-types of noise coming under the proposed taxonomy and provide examples.

Orthography: This type of noise is concerned with the way words are written. Several sub-types come under this type, which we describe below. Some of them are considered as errors, *e.g.* spelling errors, while others are looked at as variations or deviations from the standard way of writing to serve a purpose, *e.g.* word obfuscation, word lengthening.

Spelling Errors: This is when a word is spelt in a way that deviates from reference dictionaries, standardised or accepted norms, or recognised usage. The misspelling of a word takes different forms. For example, the word "receive" can be misspelt by deleting a character, *e.g.* "receve"; inserting an extra character, *e.g.* "recceive"; swapping adjacent characters, *e.g.* "recieve"; or replacing a character with another, *e.g.* "reciece" (Sakaguchi et al., 2017; Belinkov and Bisk, 2018). Additionally, a spelling error can occur when writing a word without a hyphen where needed or with a space where it should be written as one word such as writing "4 MB" instead of "4MB" (Bušta et al., 2009).

Orthographic Variants: This covers the spelling of words in different ways due to: 1) regional variations: *e.g.* British English spelling vs. American English spelling, *e.g.* "centre", "center"; 2) words with different correct spellings: *e.g.* "spelled", "spelt", "سوريا" or "سورية" for "Syria" in Arabic, or when words are transliterated, *e.g.* proper name "محمد", having one spelling in Arabic, could be written as "Muhamed", "Mohamed" or "Mohammed" in English; or 3) diacritical marks:

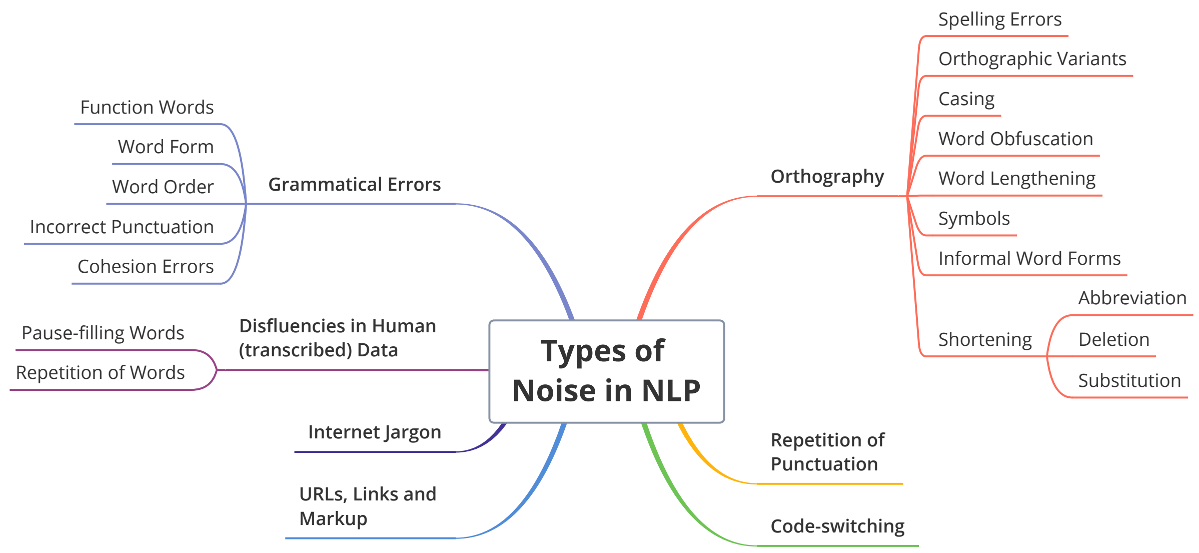

Figure 1: Types of Noise in NLP

in some languages, such as Arabic, words have accents (or what is called diacritics), which are not always used. These diacritics can change the meaning of the word. For example, the word “جد” could mean “grandfather” with the diacritical mark placed above the first letter “جَد jed” and “diligence” if this mark is placed under the first letter “جِد jid”. It is therefore important that the models are trained to recognise the difference.

Casing: This sub-type refers to instances where casing is used for a purpose. Some words are capitalised for emphasis, *e.g.* “NOTED!”. It also covers cases where casing is incorrect or missing where needed, generated by mistake and not deliberately to serve a purpose such as random capitalisation of some characters in a word, *e.g.* “SUre”; or absence of casing where needed, for example, on proper names, *e.g.* “john”.

Word Obfuscation: This sub-type refers to cases where some characters within a word are obfuscated, or in other words, disguised, using numbers or symbols. It can be used for purposes such as disguising violence, *e.g.* “ki11” instead of “kill”; or masking profanity, *e.g.* writing “fuck” as “f*ck” (Michel and Neubig, 2018).

Word Lengthening: It refers to elongating a word by replicating a letter(s) in it, often to express emphasis, *e.g.* “ Yes, Nooow!”, or sentiment, *e.g.* “قمــررر qemerrr” which means “moon” in Arabic and used to compliment a girl’s beauty.

Symbols: This covers any special symbol or sign used to express an idea, mood or feelings, *e.g.* emoticons “-:)”, emojis “😲”; or as a replacement for a word, *e.g.* using “@” to mean “at”.

Informal Word Forms: This type refers to cases where multiple words are written jointly as one word, following a certain dialectal convention (Subramaniam et al., 2009), for example, dialectal words or slang, *e.g.* “wanna” for “want to”, “whatcha” for “What are/do you ...?”. This form of contraction does not normally follow the conventions of how words are contracted and is different from other more common forms of contraction such as “isn’t” for “is not” and “aren’t” for “are not”. In the extreme case, most of the text could be written in dialect such as in Arabic (Darwish, 2014).

Shortening: This type refers to cases where a word or phrase is written in a short form using different techniques, including three sub-types:

Abbreviations includes any form of shortening of a word or phrase used to refer to the whole word or phrase, for example, “Professor” is abbreviated as “Prof”. It also covers acronyms such as Internet slang, *e.g.* “LOL” for “Laugh Out Loud”; and initials, *e.g.* “idk” for “I don’t know” (Subramaniam et al., 2009).

Deletion refers to cases where words and phrases are shortened without following any well-established patterns (*i.e.* more arbitrarily), for example, by character deletion, *e.g.* “msg” for “message”; by cutting part of a word, *i.e.* truncation, *e.g.* “tom” for “tomorrow”; or deleting an entire word, *e.g.* “drvng hm” for “I am driving home” (Subramaniam et al., 2009).

Substitution happens when words or characters are replaced with numbers or letters which have the same phonetic sound to make it shorter. Substitutions may encompass several sounds. Examples include writing “2day” for “today”, “l8r” for “later”, and “byk” for “bike” (Subramaniam et al., 2009; Gouws et al., 2011; Han and Baldwin, 2011). The use of numerals in place of letters can also happen for other proposes, *e.g.* writing Arabic text in Latin letters and using Arabic numerals to represent letters when there is no equivalent in the Latin script (Darwish, 2014). For example, the word “تحرير tahrīr” which means “liberty” could be written as “ta7rīr” with the letter “ـحـ h” being replaced with number “7” (*ibid.*). Another type of substitution errors occurring in texts includes when typing wrong keys on a keyboard instead of the intended ones (Kane et al., 2008) (this could also be seen as a spelling error, see Orthography type).

In some instances, shortening can happen using a mix of techniques, for example, by both deletion and substitution, *e.g.* “f2f” for “face-to-face”.

Grammatical Errors: This type of noise implies a deviation from the grammatical rules of a language apart from spelling errors (Lommel and Melby, 2015; Garnier and Saint-Dizier, 2016), including the following sub-types:

Function Words: Function words include prepositions, articles, determiners that are used incorrectly (Lommel and Melby, 2015), *e.g.* wrong preposition, “I bought this book **to** her” instead of “I bought this book **for** her”.

Word Form: This sub-type refers to a problem in the form of a word and includes agreement, tense-mood-aspect, and part-of-speech (Lommel and Melby, 2015), *e.g.* “I **have** a good day yesterday” (present tense) instead of “I **had** a good day yesterday” (past tense).

Word Order: This sub-type refers to instances where the order of words is incorrect (Lommel and Melby, 2015). For example, unlike in English, in Arabic, an adjective comes after a noun to describe it, so it is incorrect to say “a big house” where it must be “a house big”.

Incorrect Punctuation: Punctuation errors may include missing or incorrect placement of punctuation marks (*e.g.* !, ?, etc.) (Bušta et al., 2009). Punctuation plays a major role in our understanding of a text and text readability. For example, the sentence “Eat, dog!” could be read and interpreted differently with or without a comma.

Cohesion Errors: This type generally refers to structural errors that affect the flow of the text (within or across sentences) caused by using wrong linking words or pronouns, *e.g.* “Your car is newer, **hence** mine is faster”.

Disfluencies in Human (transcribed) Data: This type covers disfluencies that occur in spontaneous spoken language (Shriberg, 1994), including:

Pause-filling Words: This type refers to words or phrases used to express pausing in writing that mimics natural speech, *e.g.* “uh”, “er”, “um”. They generally do not have any meaning on themselves but may be indicative of important aspects, *e.g.* hesitation or surprise reaction (positive or negative), as well as style.

Repetition of Words: It refers to the occurrence of the same words several times or syntactically similar units unintentionally or on purpose, *e.g.* “**I have I have** discussed this **matter matter matter** with her again. Still, she is not convinced”.

Repetition of Punctuation: Unlike incorrect punctuation sub-type, this type refers to instances where punctuation marks such as exclamation mark or question mark are repeated to serve a purpose, *i.e.* for emphasis, *e.g.* “Really! You want me to go now???”; or to express an emotional state, *e.g.* “What???? This is really annoying!!!”.

Code-switching: This type refers to the alternation between different languages in a single sample. For example, “努力ing” is a phrase mixed with Chinese and English texts, which means “working hard”, with the English suffix “-ing” added to the Chinese word. An entire word or phrase could also be in a different language.

Internet Jargon: This type refers to new words and acronyms that gained special meaning and usage in certain social media platforms. Words such as “downvote”, “upvote”, and acronyms such as “TIL” for “Today I Learned”, “OP” for “Original Poster” are examples of jargon found on Reddit (Berard et al., 2019).

URLs, Links and Markup: This type includes web addresses and hyperlinks, *e.g.* HTML tags, URLs and Hashtags. As these elements might provide additional context, they should therefore be preserved. It also covers "@mentions" in tweets that carries textual information from the Twitter profile it refers to (Gorrell et al., 2015) and markdown special characters, *e.g.* "*" used for formatting in platforms such as Reddit (Berard et al., 2019).

4 Experimental Setup

This section describes the settings of the experiments we carried out so as to show the impact of different pre-proccessing strategies on different NLP tasks. We experiment on three tasks based on data (in English) collected from Twitter: Offensive Language Identification, Informative COVID-19 Tweets Identification, and Tweets Sentiment Analysis tasks. We start by describing each task and the data used (Section 4.1), followed by the pre-processing steps of different sources of noise (Section 4.2), and the model architectures (Section 4.3).

Due to length restrictions, we limit ourselves to experimenting with a few types of noise, with the aim to show their role as of being "harmful noise" to filter out or "useful noise" to keep, and to what extent this is dependent on the task.

4.1 Tasks and Datasets

Since it is not possible to find corpora that cover all types of noise as defined in the taxonomy, we select three datasets sourced from social media, where the texts are informal and contain several common types of noise. We use the following tasks and their respective freely available datasets (statistics in Table 1):

Offensive Language Identification (OLID) (Zampieri et al., 2019a,b): This task focuses on the problem of identifying and categorising offensive language on Social Media. We take the main type of annotation and treat the task as a binary classification task. Given the lack of standard training/development splits in the OLID dataset, we randomly split the training data into training and development sets with a ratio of 0.8/0.2. The official test set in this dataset is used for evaluation.

Informative COVID-19 Tweets Identification (COVID) (Nguyen et al., 2020): This is a binary classification task identifying whether a COVID-19 related Tweet is informative or not. We use the official train/dev/test splits in the dataset.

Twitter US Airline Sentiment Analysis (SA):[2] This consists of annotated user reviews on Twitter classified into positive, negative and neutral. We filter the data by only including the annotations where the sentiment confidence is 1, and then randomly split the data into train/dev/test sets with a ratio of 0.8/0.1/0.1.

4.2 Pre-processing

The pre-processing step is where we generally make decisions on how to deal with the data in terms of whether to clean, normalise, or keep the data as it is. We experiment with different pre-processing strategies for each task by removing, normalising or keeping seven types and sub-types of noise listed in our taxonomy: **casing, @mention tag, hashtag, emoji, code-switching, URL and punctuation**. For casing, the pre-processing involves normalising all characters to lowercase. For hashtags and emojis, pre-processing can either remove or normalise them by transforming them into corresponding word phrases that share the same semantic meaning (*e.g.* "#PutUpOrShutUp" transformed into "Put Up Or Shut Up", the emoji "😊" transformed into "smiling face"). For the other types of noise, pre-processing means removing them. In this work, we apply the same pre-processing steps to the training, development and test sets.

4.3 Model and Hyperparameters

We used a pre-trained BERT (Devlin et al., 2019) model with the "bert-base-cased" architecture[3] as our classifier. A dropout with a rate of 0.1 is applied to the output layer on top of the pre-trained model. Models were trained for 5 epochs on the training set, and the checkpoint with highest macro F1 score on the development set was selected for evaluation on the test set. We fine-tuned BERT using AdamW (Loshchilov and Hutter, 2019) optimiser with a learning rate of $1e^{-5}$. All models were trained on a single V100 GPU, with a batch size of 128 sentences. Our code was based on the

[2] https://www.kaggle.com/crowdflower/twitter-airline-sentiment

[3] We did not use the uncased BERT because it treats the uncased text the same way as its cased version, which would not allow for the comparisons by normalising/keeping the casing information.

	OLID		COVID		SA		
	OFF	NOT	INFOR	UNINF	POS	NEG	NEU
Train	3,518	7,074	3,273	3,663	1,212	5,906	1,238
Dev	882	1,766	472	528	152	738	155
Test	240	620	944	1,056	151	738	155

Table 1: Number of sentences in the three dataset. For the sentiment analysis data, we report the statistics after filtering. OFF: label "offensive". NOT: label "non-offensive". INFOR: label "informative". UNINF: label "uninformative". POS: label "positive". NEG: label "negative". NEU: label "neutral".

huggingface[4] BERT implementation. We ran each training with three random seeds and reported the averaged of the test macro F1 score.

5 Results

We first present our baseline performance without any pre-processing in Table 2. We then show the percentage change in macro-F1 scores for each of the three tasks by addressing one type of noise from the baseline at a time in Figure 2. Each time one type of noise is removed in the pre-processing except the casing, which is lowercased. We consider noise to be "useful" noise when the removal results in decrease in performance (negative bars). It is worth noting that if the performance drops, that means removing the noise might deviate the intended meaning of the original sentence.

	OLID	COVID	SA
baseline	0.793 ± 0.009	0.867 ± 0.002	0.853 ± 0.026

Table 2: Baseline results in macro-F1 scores on the three tasks.

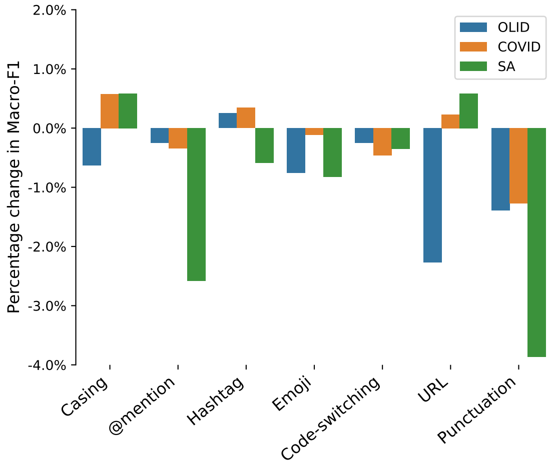

Figure 2: Percentage change in macro-F1 score with the baseline (no pre-processing) when removing one type of noise at a time.

Comparing the three tasks, removing/normalising the same type of noise clearly has opposite effects (*e.g.* casing), or different magnitudes of effect (*e.g.* punctuation):

Normalising **casing** leads to a decrease on OLID task whereas the performance on the other two tasks increases. This is intuitive as offensive texts are likely to be written in uppercase while it might be less useful for identifying informative tweets or sentiment (*i.e.* for sentiment, when both strong positive and strong negative texts involve uppercasing, the casing information does not indicate sentiment correctly). A similar trend can be witnessed when removing **URLs**, but the decrease on OLID task is larger. This might be because, in the OLID dataset, web addresses have already been normalised with a unified "URL" token, thus it might contain useful rather than noisy information. We can therefore say that casing and URLs are useful noise and should be kept on OLID task for better performance.

Hashtags are more useful on the sentiment analysis task because, when removed, the performance drops on the SA task while improves on the other two tasks. We notice that in the data for sentiment analysis, hashtags are mostly single words such as "#mad" and "#senseless" so that the sentiment could be detected by the model. However, in the OLID dataset, hashtags hinting toxicity mostly include multiple words, *e.g.* "#Liberalismisamentaldisorder", which increases the difficulty of utilising these hashtags. In the COVID dataset, almost all tweets consist of hashtags related to COVID-19, thus the hashtags do not help identifying whether the tweet is informative or not.

Removing **@mentions** causes a more obvious decrease in the SA task than the other two tasks. This is because the @mentions in OLID and COVID datasets have been normalised with a unified "@USER" token, but in the SA data @mentions stay in the form of usernames. We found that the @mentions in sentiment analysis data helps indicate the sentiment. For example, 32.0% of sentences with "@VirginAmerica" are labeled as

[4] https://github.com/huggingface/transformers

positive while there are only 10.9% positive tweets with “@united”. Similarly, removing **emojis**, **code-switching** and **punctuation** leads to a decrease on all three tasks. However, emojis are less influential for identifying informative tweets. The decrease therefore is not significant for this specific task. Furthermore, as the non-English words are mostly named entities, the removal of code-switching could break the sentence structure. Regarding punctuation, it is important for the three tasks where its removal causes larger performance drop especially on the SA task as it leads to the removal of emoticons “:-)”, which can be useful for classifying sentiment.

Based on our findings that show how different strategies lead to different results, we took a step further to show the validity of our reasoning for how noise should be understood and handled on different NLP tasks. To that end, we combined the different pre-processing strategies and trained two other systems: **remove all**, which does the lowercasing and removes all other types of “noise” we dealt with in our experiment (*i.e.* URLs, hashtags, @mentions, emojis, code-switching and punctuation), and **remove+keep**, which only removes the harmful noise in the specific task as showed in Figure 2 for each type of “noise” (*e.g.* for OLID task only hashtag is removed while for SA task, URL is removed and the texts are lowercased). In addition, to make use of the potentially useful information in hashtags and emojis, we followed the state-of-the-art approaches (Liu et al., 2019; Kumar and Singh, 2020) and segmented hashtags into separate words and transformed emojis into corresponding English phrases as we stated in Section 4.2 (pre-processing of other types of noise is the same as **remove+keep**). The system trained on this data is noted as **remove+keep+transform**. The results are presented in Figure 3.

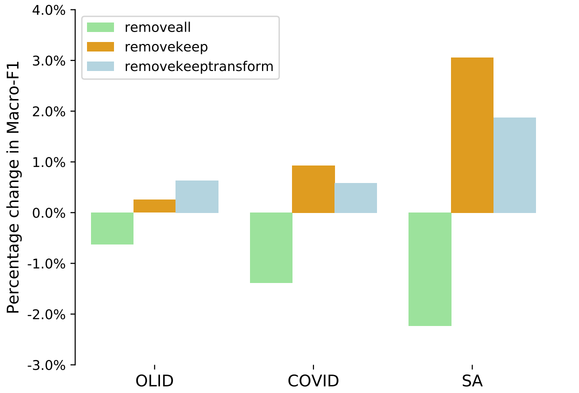

Figure 3: Percentage change in macro-F1 score with the baseline when removing all/removing harmful and keeping useful noise/SOTA pre-processing. The pattern indicates that the improvement/decrease is statistically significant ($p < 0.05$).

The system with data removing all sources of noise in the pre-processing shows a poorer performance than the baseline, which keeps all sources of noise. However, both “remove+keep” and “remove+keep+transform” systems outperform the baseline, with improvement on sentiment analysis task being the most significant. However, after transforming hashtags and emojis into English phrases, the performance only improves on the OLID task compared to the “remove+keep” system, which confirms our claim that the same noise normalisation strategy to noise might have different influence on different tasks.

These results are in line with our proposed definition of noise in NLP where some types of noise can be useful and others harmful, and how this is greatly dependent on the task. They also confirm our suggestion that one should not simply follow “standard” pre-processing pipelines, but carefully devise appropriate strategies to deal with different types of noise depending on the task.

6 Conclusions

In this paper, we proposed a definition and taxonomy of noise in NLP so as to serve as a point of reference for NLP researchers to consult when they devise strategies to clean, normalise, or embrace non-standard content at either training or inference time to improve the robustness of their systems to this unseen or unexpected naturally occurring harmful and useful noise. We highlighted that noise in NLP should be carefully handled in light of what we call “harmful noise” that needs to be removed when it affects the performance of the system and/or it does not carry the intended meaning of the text, and “useful noise” that needs to be kept because it is an integral part of the data and useful for a task, or even should be added to the training data when it only happens naturally at test time.

Our experiments support our argument by demonstrating that tailored approaches are better than blanket, all-encompassing solutions generally applied by researchers through “standard” pre-processing pipelines. For instance, we found out that casing and URLs are useful noise and should be kept on OLID task, but having a negative impact on the other two tasks (SA and COVID tasks) and should therefore be removed. We have also shown how special handling of harmful and useful noise could result in better performance where remove-all and keep-all approaches resulted in poorer performance. Our approach to noise was based on their impact on the tasks - we removed types of noise which had negative influence on the tasks. Our goals were to bring awareness to the different types of unexpected content and provide a definition and a taxonomy, and to highlight the fact that they need to be handled carefully rather than being avoided or treated using the same strategies.

References

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, Beijing, China. Association for Computational Linguistics.

Kanika Bansal and Rajiv Kumar. 2013. K-algorithm: A modified technique for noise removal in handwritten documents. *International Journal of Information Sciences and Techniques*, 3(3):1–8.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019. Naver labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.

Jan Bušta, Dana Hlavácková, Miloš Jakubícek, and Karel Pala. 2009. Classification of errors in text. *RASLAN 2009 Recent Advances in Slavonic Natural Language Processing*, pages 109–119.

Kareem Darwish. 2014. Arabizi detection and conversion to Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 217–224, Doha, Qatar. Association for Computational Linguistics.

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.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia. Association for Computational Linguistics.

Marie Garnier and Patrick Saint-Dizier. 2016. Error typology and remediation strategies for requirements written in English by non-native speakers. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 190–197, Portorož, Slovenia. European Language Resources Association (ELRA).

Genevieve Gorrell, Johann Petrak, and Kalina Bontcheva. 2015. Using @twitter conventions to improve #lod-based named entity disambiguation. In *The Semantic Web. Latest Advances and New Domains*, pages 171–186, Cham. Springer International Publishing.

Stephan Gouws, Dirk Hovy, and Donald Metzler. 2011. Unsupervised mining of lexical variants from noisy text. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 82–90, Edinburgh, Scotland. Association for Computational Linguistics.

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*, pages 368–378, Portland, Oregon, USA. Association for Computational Linguistics.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. 2018. How robust are character-based word embeddings in tagging and MT against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 68–80, Boston, MA. Association for Machine Translation in the Americas.

Shaun K Kane, Jacob O Wobbrock, Mark Harniss, and Kurt L Johnson. 2008. Truekeys: identifying and correcting typing errors for people with motor impairments. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 349–352.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Priyanshu Kumar and Aadarsh Singh. 2020. NutCracker at WNUT-2020 task 2: Robustly identifying informative COVID-19 tweets using ensembling and adversarial training. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 404–408, Online. Association for Computational Linguistics.

Jinyu Li, Li Deng, Reinhold Haeb-Umbach, and Yifan Gong. 2015. *Robust automatic speech recognition: a bridge to practical applications*. Academic Press.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ariel Lommel and AK Melby. 2015. Multidimensional quality metrics (mqm) - issue types. `"https://www.qt21.eu/mqm-definition/definition-2015-06-16.html#issue_types"`. Accessed: 18/08/2021.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. 2020. WNUT-2020 task 2: Identification of informative COVID-19 English tweets. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 314–318, Online. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3281–3287. AAAI Press.

Elizabeth Ellen Shriberg. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. thesis, University of California, Berkeley, CA.

L. Venkata Subramaniam, Shourya Roy, Tanveer A. Faruquie, and Sumit Negi. 2009. A survey of types of text noise and techniques to handle noisy text. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, AND '09, page 115–122, New York, NY, USA. Association for Computing Machinery.

Kaveh Taghipour, Shahram Khadivi, and Jia Xu. 2011. Parallel corpus refinement as an outlier detection algorithm. *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 414–421.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. 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 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiao Xiong. 2009. *Robust speech features and acoustic models for speech recognition*. Ph.D. thesis, Nanyang Technological University.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. 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 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.