Natural Language Processing (2026), 1–47
doi:10.1017/nlp.2026.10020

CAMBRIDGE UNIVERSITY PRESS

SURVEY PAPER

Misspellings in natural language processing: A survey of recent literature

Gianluca Sperduti and Alejandro Moreo

Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Italy
Corresponding author: Gianluca Sperduti; Email: gianluca.sperduti@isti.cnr.it

(Received 27 January 2025; revised 5 January 2026; accepted 11 February 2026)

Abstract

This survey provides an overview of the challenges of misspellings in natural language processing (NLP). Misspellings are ubiquitous in digital communication, and even if humans can generally interpret misspelt text, NLP models frequently struggle to handle it: this causes a decline in performance in common tasks like text classification and machine translation. In this paper, we reconstruct a history of misspellings as a scientific problem. We then discuss the latest advancements to address the challenge of misspellings in NLP. Main strategies to mitigate the effect of misspellings include data augmentation, double step, character-order agnostic, and tuple-based methods, among others. This survey also examines dedicated data challenges and competitions to spur progress in the field. Critical safety and ethical concerns are also examined, for example, the voluntary use of misspellings to inject malicious messages and hate speech on social networks. The survey also explores psycholinguistic perspectives on how humans process misspellings, potentially informing innovative computational techniques for text normalisation and representation. Additionally, the survey explores the challenges that misspellings pose in multilingual contexts. Finally, the misspelling-related challenges and opportunities associated with modern large language models are also analysed, including benchmarks, datasets and performances of the most prominent language models against misspellings. This survey provides a comprehensive review of recent research on misspellings and aims to serve as a valuable resource for researchers seeking to get up to speed on this problem within the rapidly evolving landscape of NLP.

Keywords: Misspellings; text normalisation; user-generated content; data augmentation; hate speech detection

1. Introduction

Human language is constantly evolving. The world we live in is governed by information and communication technologies. Our time, sometimes dubbed *the digital era*, must thus be prepared to face changes in the way we communicate, and implement mechanisms to adapt to it.

Changes in communication derive from multiple aspects. The use of non-standard written language might stem from cultural or societal factors, among others, or it may simply happen by *mistake*. In this survey, we refer to both phenomena as *misspellings*. Misspellings have become pervasive in the digital written production since the revolutionary Web 2.0 led people interact freely through social media, blogs, forums, etc. Even though misspellings are generally unintentional, in some contexts, these may also be *intentional*. Of course, the presence of misspellings complicates the reading of a text. Notwithstanding this, and somehow, surprisingly, humans have the ability to read and comprehend misspelt text without much effort and, sometimes, even without realising their presence (Andrews 1996; Healy 1976; McCusker *et al*. 1981; Shook *et al*. 2012). Computers do not have similar capabilities, though. Although the NLP community has long downplayed the

problem of misspellings (if not for grammatical error correction (Shishibori *et al.* 2002) or text normalisation (Damerau 1964)), it is by now abundant evidence that misspellings represent a serious risk to the performance of NLP systems (Baldwin *et al.* 2013; Heigold *et al.* 2018; Moradi and Samwald 2021; Náplava *et al.* 2021; Nguyen and Grieve 2020; Plank 2016; Vinciarelli 2005; Yang and Gao 2019), even for the latest generation of large language models (Moffett and Dhingra 2025).

This survey addresses misspellings as a pervasive phenomenon that negatively impacts downstream NLP tasks. We do not focus on general-purpose automatic spelling correction methods, for which recent, comprehensive reference material is already available (see, e.g., Bryant *et al.* 2023; Hládek *et al.* 2020; Wang *et al.* 2021b). Instead, our focus is on the implications and challenges that misspellings pose for NLP methods that are explicitly designed to be robust to them in downstream applications. We cover research published since 2009, as earlier work is already comprehensively reviewed by Subramaniam *et al.* (2009). Since then, the topic has gained increasing attention. A growing number of methods have been proposed to specifically address the problem of misspellings (Belinkov and Bisk 2018; Heigold *et al.* 2018), alongside dedicated benchmarks (Michel and Neubig 2018) and even shared tasks and data challenges (Basili *et al.* 2010; Dey *et al.* 2011; Lopresti *et al.* 2009). This survey aims to provide a comprehensive overview of these recent advancements in the field.

The study of misspellings in NLP is paramount not only as a means for improving the performance of current systems, but also for reasons that are ultimately bound to *safety* and *ethics*. Misspellings are, as hinted above, not always an involuntary phenomenon. Misspellings may sometimes be carefully and maliciously designed (Li *et al.* 2019a) with the purpose of disguising certain words to elude the control of automatic content moderation tools or spam detection filters. The study of misspelling can help in mitigating the proliferation of hate speech or in preventing unwanted content from reaching the final audience. Additionally, the fact that certain misspellings act as obfuscations for computers but not for humans suggests that studying this phenomenon from a psycholinguistic perspective might inspire alternative, more efficient methods for text representation and processing.

The rest of this survey is organised as follows. In Section 2, we offer an overview of the history of misspellings in the digital era, analysing the main trends before and after the proliferation of user-generated content, the upsurge of neural networks and the advent of large language models (LLMs). In Section 3, we describe how the phenomenon is regarded through the lens of linguistics and NLP. In Section 4, we survey previous work on the potential harm of misspellings. Section 5 is devoted to describing methods specifically devised to counter misspellings. Section 6 deals with the challenges misspellings pose in multilingual contexts. The main tasks, evaluation measures, venues and datasets dedicated to misspellings are discussed in Section 7. Section 8 is devoted to analysing the phenomenon of misspellings from the point of view of modern LLMs. Section 9 discusses the main applications in which the presence of misspellings gains special relevance. Section 10 concludes by also pointing to promising directions of research.

2. A brief history of misspellings

The history of misspellings in NLP spans several decades, dating back at least to Blair (1960); Damerau (1964) seminal works on spelling error detection and correction published in the 1960s. Here, we provide a concise overview of this long-standing topic, focusing on three main phases that hinge upon the proliferation of the so-called *Web 2.0* and the subsequent spread of (often carelessly generated) user-generated content, and the advent of LLMs. The term *Web 2.0* was first coined by Darcy DiNucci in 1999, but it was not until 2004 that it gained popularity through the

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Web 2.0 Conference.[1] It took some time for user-generated content to take hold on the Internet, something we identify as happening around 2010. This section, therefore, briefly surveys the history of misspellings before (Section 2.1) and after (Section 2.2) this turning point.

2.1 Before 2010: fewer data, less misspellings

Before the explosion of user-generated data on the Internet, the vast majority of content available on the web (static web pages, journal articles, etc.) was characterised by the fact that the content was moderately well curated. As a result, the amount of data was relatively limited, and the available data contained few misspellings. For this reason, automated text analysis technologies were rarely concerned with the presence of misspellings, if at all. The study of misspellings was confined to the development of automatic correction tools that aid users in producing misspelling-free texts by, for example, correcting typos or applying OCR-produced errors.

Arguably, the first works on misspellings were those by Blair (1960) and Damerau (1964), which proposed the earliest dictionary-based methods for spelling correction. In this survey, we do not focus on spelling correction *per se* (we refer the interested reader to Bryant *et al.* 2023; Hládek *et al.* 2020; Wang *et al.* 2021b), but rather on NLP systems designed to be resilient to misspellings. In this context, Vinciarelli (2005) stands out as one of the pioneering studies, specifically addressing errors introduced by OCR technologies.

Some studies seemed to indicate that the problem of misspellings was not paramount for text classification technologies, at least when these concern the classification by topic of (curated) text documents (Agarwal *et al.* 2007). The situation differed somewhat when shifting to other, less curated sources, such as emails, blogs, forums and SMS data, or when analysing the output generated by automatic speech recognition engines from call centres. The problem attracted little attention from the research community at the time, and it was not until 2007 that a dedicated workshop, called *Analysis for Noisy Unstructured Text Data* (AND), emerged and renewed interest in the field (see also Section 7.4).

To the best of our knowledge, the only survey on NLP systems robust to noise was published in 2009 (Subramaniam *et al.* 2009). This survey primarily focused on handling noise in OCR scans, blogs, call centre transcriptions and similar sources.

2.2 After 2010: the rise of social networks and deep learning

Since 2010, user-generated content has become increasingly pervasive, mainly due to the revolution of social networks. At the same time, deep learning technologies have taken the world by storm (Krizhevsky *et al.* 2012), not only due to the increase in performance they show off in most NLP tasks (Collobert *et al.* 2011), but also because of their potential to eliminate the need for manual feature engineering; instead, the neural network itself learns to represent the input. This raises questions about the necessity of a pre-processing step for correcting misspellings beforehand.

The increasing prevalence of misspelt data and the proliferation of NLP technologies have inspired numerous studies analysing the impact of misspellings on state-of-the-art models (Section 4), as well as papers proposing systems that are resilient to misspellings (Section 5).

The study of misspellings in NLP presents significant benefits. The most apparent advantage is the enhancement of performance in any NLP tool, but not only. Systems that are resilient to misspellings are also *safer*. For reasons discussed later, some misspellings are *intentional*, designed to evade the scrutiny of content moderation tools or spam filters. Ultimately, the study of misspelling resilience aims to deepen our understanding of language (further discussions are offered in Section 10).

[1] Source: https://en.wikipedia.org/wiki/Web_2.0.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

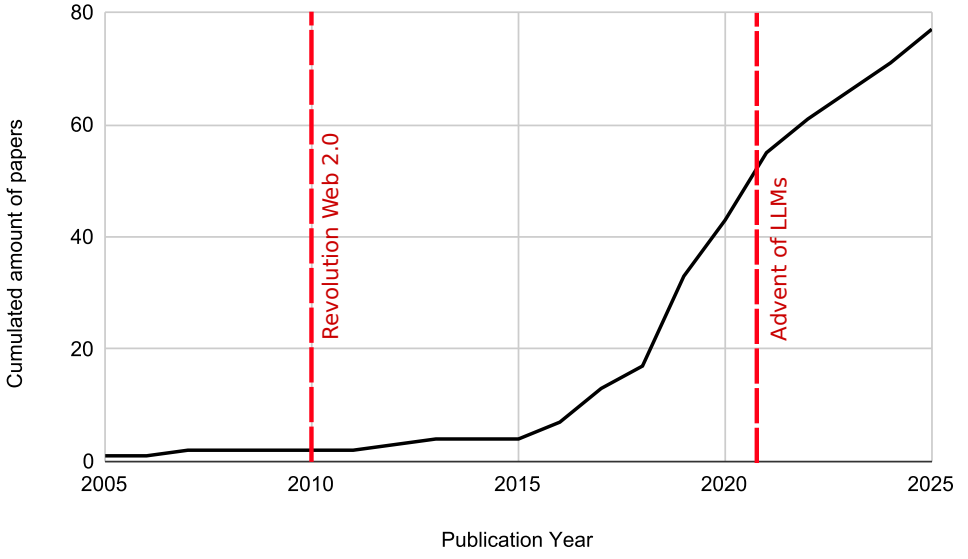

Figure 1. Publication trends in NLP papers on misspellings (2004–2025).

This increased interest in the subject was partially fostered by the work of Belinkov and Bisk (2018); Heigold *et al.* (2018); Edizel *et al.* (2019), who showed the performance of different models decrease noticeably in the presence of misspellings. This renewed momentum has led to the appearance of dedicated workshops devoted to studying the phenomenon from the point of view of user-generated content (such as the WNUT workshop series[2]) as well as from the point of view of machine translation (such as the WMT workshop/conference series[3]); more information about dedicated tasks and datasets can be found in Section 7.4.

Between 2017 and 2022, neural machine translation emerged as the most prolific field in the study of misspellings, closely followed by sentiment analysis.

2.3 After 2020: the advent of LLMs

In recent years, the trend has shifted markedly: whereas earlier research focused heavily on developing methods to combat misspellings in downstream tasks (with a disproportionate emphasis on machine translation), the field has now moved toward a growing number of *evaluation* papers that analyse the ability of LLMs to handle misspellings.

The advent of LLMs around 2020–2021 has dramatically reshaped the NLP and AI landscape, including research on misspelling resilience. Due to their high computational costs, LLMs are not easily amenable to experimentation with new training methods aimed at improving robustness to misspellings. Moreover, some language models (especially commercial ones) already exhibit strong resistance to misspellings (Moffett and Dhingra 2025). Nevertheless, there is substantial interest in understanding how LLMs handle misspellings in both downstream tasks and more general settings, as evidenced by the relatively high number of papers published on the topic between 2024 and 2025 (see, e.g., Pan *et al.* 2024; Wang *et al.* 2024, 2025; Zhang *et al.* 2024).

Figure 1 shows the publication trends with respect to all NLP papers related to misspellings.

[2] https://aclanthology.org/venues/wnut/.
[3] https://aclanthology.org/venues/wmt/.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

3. What is a misspelling?

The term *misspelling* is too broad a concept, which has come to encompass many different types of unconventional typographical alterations. In this section, we turn to review the main considerations behind this term as viewed through the lens of *linguistics* and *NLP*, and we try to break down the many subtle nuances it encompasses. While in NLP, the terms *misspelling* and *noise* are by and large interchangeable, in linguistics, the term *error* is more commonly employed. Other terms like *typo*, *mistake* or *slip* are often used in more general, non-specialised contexts. Throughout this survey, we prefer the term *misspelling* since it clearly evokes a link with text, and since *noise* and *error* are too wide hypernyms that the target audience of this survey might find rather ambiguous.

3.1 Under the lens of linguistics

In linguistics, there are primarily three fundamental viewpoints for models dealing with misspellings, which we cover in this section. The first is that of *general linguistics* (Section 3.1.1), where some authors have attempted to define and categorise various types of misspellings. The second one is that of *sociolinguistics* (Section 3.1.2), in which authors analyse misspellings from a social perspective. The last one originates from *psycholinguistics* (Section 3.1.3), which rather focuses on cognitively relevant aspects of the problem.

3.1.1 General linguistic perspective

As already mentioned, in general linguistics, there is no broadly agreed-upon definition of what a *misspelling* is, and the term *error* is often preferred. Error analysis is one important field of linguistics that studies the phenomenon of errors in second language learners. James (2013) defines errors as *an unsuccessful bit of language*. Richards and Schmidt (2013) instead define errors as the use of a linguistic item (e.g., a word, a grammatical unit, a speech act, etc.) in a way that a fluent or native speaker of the language regards as showing faulty or incomplete learning.

In general linguistics, it is customary to draw a distinction between *error* and *mistake*. Richards and Schmidt (2013) point out that errors are due to a lack of knowledge of the speaker, while mistakes are made because of other compounding reasons, such as fatigue or carelessness. In the same work, errors are classified as belonging to *lexical error* (i.e., surface forms which are not included in a vocabulary), *phonological error* (i.e., in the pronunciation) and *grammatical error* (i.e., not compliant to syntactic rules). Interestingly enough, none of these concepts seems to embrace the possibility that misspellings may be created as a voluntary act (more on this later).

3.1.2 Sociolinguistic perspective

Sociolinguistics focuses on the concept of *spelling variation* (Nguyen and Grieve 2020). The word *misspelling* itself carries an implicit judgement against the author: the author of the noise is responsible for *missing* the correct normative spelling of the word. Contrarily, the sociolinguistic perspective considers that there are no such errors, but rather *variations* in spelling. These variations can originate from social needs, such as avoiding censorship, expressing group identity or representing regional or national dialects.

Over time, linguistic koinés and speech communities may develop alternative spellings for certain words, whether intentionally or through gradual convention. In the NLP literature, this phenomenon is also referred to as *spelling variation*, that is, a deviation from the standard orthography that may or may not be perceived as an error. Since there is no universally agreed definition of *misspelling*, these socially driven orthographic variations deserve special attention. From a morphological standpoint, they can be seen as systematic deviations from the normative form, and as such they evolve diachronically, alongside the language itself.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

3.1.3 Psycholinguistic perspective

After outlining key conventions in general linguistics and sociolinguistics, it is essential to emphasise the significant relationship between psycholinguistics and the phenomenon of misspellings. As stated by Fernández and Cairns (2010), psycholinguistics investigates the cognitive processes involved in the use of language, rather than the structure of language itself. In the case of reading, psycholinguistics is concerned with understanding the cognitive processes that underlie this activity, from the acquisition of the sensory stimulus derived from the visual perception of letters, to the subsequent comprehension and cognitive reorganisation of the information within the brain.

The branch of psycholinguistics that is most relevant to the topic of this survey is the one devoted to studying the cognitive processes behind the acts of writing and reading. It has been noted on several occasions that humans are able to read long and complex sentences that include misspelt words with little reduction in performance. The most notable example of this is that of *garbled words*, in which the internal letters are randomly transposed. Despite this, humans are able to read them with high accuracy. Some related work in the psycholinguistics literature includes the work by Andrews (1996); Healy (1976); McCusker *et al.* (1981); Shook *et al.* (2012). This cognitive ability of human beings has inspired some of the methods we describe in Section 5.2.

Some researchers in the field of NLP have gained inspiration from lessons learnt in psycholinguistics and have taken advantage of these to devise models robust to misspellings. For example, characters that are graphically similar can be interchanged without significantly affecting human reading comprehension (e.g., *cl0sed* for *closed*). This other intuition has inspired some of the methods that we discuss in Section 5.5.

From a computational point of view, the study of misspellings would certainly benefit from the synergies with linguistics and psycholinguistics. The cognitive abilities humans display represent a source of inspiration for methods dealing with misspellings or the creation of adversarial attacks. As an example, consider spam emails in which the content is made of garbled words or in which graphically similar characters have been replaced.

3.2 Under the lens of NLP

In the field of NLP, there is no single, clear-cut definition of misspelling. Indeed, the same type of problem (morphological error) is often expressed with different words, such as *noise*, *typo*, and *spelling mistake*.

The most common of these, along with *misspelling*, is *noise*, which is defined as any non-standard spelling variation (Nguyen and Grieve 2020). While this definition of noise may seem appropriate, any attempt to provide a universal definition of misspellings would appear contrived and, above all, imprecise.

To tackle this issue and establish clear boundaries around the concept of misspelling, NLP researchers have proposed various categorisations, which draw from different points of view: For example, from the perspective of the word surface, from the point of view of the user who generated them, or pointing up to the methods used to generate misspelt datasets in an experimental setting. The types of misspellings can be fine-grained, where multiple categories of misspellings are defined in detail, or less fine-grained, where fewer, more representative categories are selected. This lack of uniformity, among other things, complicates the search for relevant papers on the subject (which has indeed represented one of the significant challenges we faced when developing this survey).

In this section, we cover some of the main categorisations of misspellings proposed in the literature. In doing so, we note that the problem of misspelling can be approached from diverse viewpoints and thus there are multiple perspectives on this matter. For example, Heigold *et al.* (2018) have established three categories based on the **word surface**:

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

- character swaps: *nice* → *ncie*, the position of two subsequent characters is exchanged;
- word scrambling: *absolute* → *alusobte*, the order of the characters is permuted with the exception of the first and last one;
- character flipping: *nice* → *nite*, one character is replaced by another.

Belinkov and Bisk (2018) proposed an alternative classification of misspellings into two main categories based on the **dataset generation method**:

- Natural misspellings: real misspellings that occur in real-world data spontaneously;
- Synthetic misspellings: misspellings artificially generated by means of procedural rules.

Note that the differentiation between *natural* and *synthetic* misspellings does not establish a clear-cut boundary, as any misspelling generated synthetically could plausibly occur naturally. That is to say, the difference between natural and synthetic misspellings is extrinsic to the word surface form, and regards the mode of generation (spontaneous *vs.* procedural) while both represent (or mimic) the same underlying phenomenon. Indeed, this distinction is functional to experimental setups and was originally conceived with dataset generation in mind. Synthetic misspellings are more widely used due to the low frequency of natural misspellings, which hinders the collection of large, diverse corpora.[4]

What Belinkov and Bisk (2018) referred to as *natural* misspellings actually involves lexical lists that provide context for the misspellings, which can be exploited to artificially introduce natural-sounding misspellings into otherwise correct sentences. In such cases, we will refer to a third category of *hybrid* misspelling, and reserve the term *natural* misspelling for real-world misspellings found in actual data. Further aspects related to dataset generation will be discussed in Section 7.4.

Several other researchers, including van der Goot *et al.* (2018) and Nguyen and Grieve (2020), have emphasised the *user's intention* rather than focusing solely on the surface-level characteristics of misspelt words. They propose to distinguish between *intentional* and *unintentional* misspellings. For instance, Nguyen and Grieve (2020) observed that intentional misspellings, such as lengthening a word, are often used to add emphasis to an opinionated statement. An example of this could be the use of the interjection *wow* in a context where the user wants to emphasise their surprise, as in *wooooooooooow*. Furthermore, the categorisation of misspellings can be more or less *fine-grained*; in this respect, van der Goot *et al.* (2018) have proposed as many as 14 different categories of misspellings, while Heigold *et al.* (2018) have proposed only 3.

It is thus important to bear in mind that the variability in the terminologies and approaches to this topic—along with the lack of a universal definition for this type of problem—represents one of the greatest challenges faced by NLP researchers. As for our survey, we found it particularly useful to list the types of misspellings for dataset generation (see Section 7.4).

4. How serious is the problem?

SC/TN tools are commonly employed as a pre-processing step in many industrial applications as a way to cope with misspellings, while the core of the system is designed to work with clean text. (Such an approach represents the simplest scenario within the so-called *double-step methods* that will be surveyed in Section 5.3.) A legitimate question that arises is the following: *Can we simply rely on SC/TN tools and consider the problem solved?*

As the reader might have wondered, the answer is *no*. According to Plank (2016), non-standard (or non-canonical) language is a complex matter: there is no commonly agreed definition of what

[4] There are no generally agreed statistics of misspelling rates in real texts, since such a datum would largely depend on the domain of the texts (Baldwin *et al.* 2013) as well as on the definition of misspelling itself.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

constitutes a misspelling, nor of what makes a text be considered *normalised*. For instance, Plank (2016) notes that there is no single standard form for spelling variations such as *labor* in American English and *labour* in British English. This highlights the fact that the notion of a misspelling is, in some cases, inherently context-dependent, that is, what counts as an error in one variety (e.g., *labour* in American English) may be fully standard in another (e.g., British English), and that NLP systems need to account for such contextual dependencies on a task-by-task basis (more about context-dependent misspellings can be found in Mays et al. 1991).

From an application-oriented perspective, the use of spelling correction and text normalisation tools is often beneficial, as it can improve downstream performance by removing orthographic noise safely in many contexts. However, from a linguistic research perspective, and for certain specific applications (Section 9), such tools may mask valuable information about variation, evolution or intentional deviations from standard orthography. For example, text normalisation may remove dialectal traits that could be essential for certain analyses. In literary contexts, for instance, preserving a character's role might require translating a dialectal expression in the source language into a comparable dialectal expression in the target language. Finally, psycholinguistics studies suggest humans are capable of processing misspellings without significant effort (Andrews 1996; McCusker *et al.* 1981; Rayner *et al.* 2006). By removing misspellings as a pre-processing step, we lose the opportunity to better comprehend how natural language is processed and how to improve automated NLP tools accordingly.

A second, legitimate question is *Can we simply ignore the phenomenon?* This section is devoted to answering this question. Throughout it, we offer a comprehensive review of past efforts devoted to quantifying the extent to which vanilla systems' performance degrades when in the presence of misspellings. This performance decay is typically large, and is typically assessed with respect to artificial and natural misspellings (Baldwin *et al.* 2013; Belinkov and Bisk 2018).

Note that the work presented in this section focuses on measuring the impact of misspellings in methods that do not make any attempt to counter them. Systems specifically designed to be robust against misspellings will be described in Section 5.

4.1 The harm of synthetic misspellings

Synthetic misspellings are the most commonly employed type of misspelling in the related literature, likely due to the ease with which they can be artificially generated, making them convenient for testing specific approaches without the need for large datasets of naturally occurring errors.

In this section, we review related studies, dividing them into two groups: those conducted before the transformer era (Section 4.1.1) and those focusing on quantifying the impact of misspellings on BERT and other transformer-based models (Section 4.1.2).

4.1.1 Impact on pre-BERT models

The problem was partially dismissed by Agarwal *et al.* (2007), who tested traditional classifiers (SVM and Naive-Bayes) using bag-of-words representations, against misspellings generated using an automatic tool (dubbed *SpellMess*) which considers insertions, deletions, substitutions and QWERTY errors,[5] in two well-known datasets for text classification (Reuters-21578 and 20 Newsgroups). Their results show that even moderately high levels of noise (affecting up to 40 per cent of the words) did not affect classification accuracy as much as expected. The authors conjectured that this can be explained by the fact that many of the features affected by noise are rather uninformative, and that when classifying by topic, abundant patterns still remain in the rest of the training data, even at high levels of noise.

[5] Character substitutions are governed by the proximity of the keys in a QWERTY layout, see Section 7.4.2.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Quite some time later, Belinkov and Bisk (2018) confronted various Character-based and BPE-based[6] encoded neural translators with synthetic misspellings. Their results demonstrated that all machine translation models were significantly affected by the presence of synthetic misspellings.[7] This paper became influential and has served to raise awareness on the problem of misspellings.

Inspired by the latter, Naik *et al.* (2018) conducted robustness experiments on Natural Language Inference (NLI) models using different types of synthetic misspellings, such as swapping adjacent characters or inserting QWERTY errors. The authors designed a stress test to assess whether the qualitative results of NLI models are driven not only by strong pattern matching but also by genuine natural language understanding procedures. The paper goes on by demonstrating that the performance of NLI models, built on top of BiLSTMs and Word2Vec, declines when misspellings are inserted in the test set.

Heigold *et al.* (2018) carried out experiments considering different types of synthetic noise on the tasks of morphological tagging and machine translation and using different types of encodings, including word-based, Character-based, and BPE-based ones. In their experiments, different models were trained independently on different variants of the training set, including *clean*, the original set without misspellings; *scramble*, obtained by permuting the order of the characters with the exception of the first and last one in each word; *swap@10*, that randomly swaps 10 per cent of subsequent characters; and *flip@10*, that randomly replaces 10 per cent of the characters with another one. Every pair of (system, training set variant) was tested against similar variants generated out of the test set. The results show the performance of all tested models degrades noticeably when exposed to synthetic misspellings different from those on which the system was trained.

4.1.2 Impact on BERT and transformer-based models

BERT, the popular transformer model proposed by Devlin *et al.* (2019), has garnered a great deal of attention due to its ability to deliver state-of-the-art performance across a wide range of NLP tasks. Given its success, several studies have focused on analysing the sensitivity of BERT-based models to misspellings. Yang and Gao (2019) tested Vanilla BERT (i.e., a raw instance of BERT that does not implement any specific method to counter misspellings) in both extrinsic (customer review and question answering) and intrinsic (semantic similarity) tasks. To this aim, the authors employed both word-level noise (i.e., noise involving the addition or removal of entire words from a sentence) as well as various types of misspellings, showing that misspellings are significantly more detrimental to BERT than word-level noise.

Kumar *et al.* (2020) confronted BERT against QWERTY misspellings at various probabilities. The scope of this work was to quantify the extent to which the presence of misspellings harms the performance of a fine-tuned BERT in the tasks of sentiment analysis (on IMDb and SST-2 datasets) and textual similarity (STS-B dataset). Their results demonstrated that BERT is highly sensitive to this type of misspelling, even at low rates.

Moradi and Samwald (2021) carried out a systematic evaluation of BERT and other language models (RoBERTa, XLNet, ELMo) in different tasks (text classification, sentiment analysis, named-entities recognition, semantic similarity, and question answering), considering different types of synthetic misspellings. Their results confirm that the performance of all tested models degrades noticeably for all tasks and types of misspelling. For instance, RoBERTa experiences a significant decrease in performance, with a loss of 33 per cent accuracy in text classification and a 30 per cent decrease in accuracy in sentiment analysis.

Ravichander *et al.* (2021) conducted an evaluation assessment of the effect of misspellings on question-answering performance, considering various state-of-the-art language models (BiDAF,

[6] *Byte Pair Encoding* (BPE) is an encoding method operating at the subword level. Pairs of tokens that appear together frequently are grouped together and encoded using a new token.

[7] In their experiments, Belinkov and Bisk (2018) also considered *hybrid* misspellings (these are discussed later in Section 4.2), showing the harm of synthetic misspellings to be more serious than that caused by hybrid misspellings.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

BiDAF-ELMo, BERT, and RoBERTa). The authors of this study focused on errors induced by specific input interfaces (such as translation, audio transcription, or keyboard) and devised ways for generating synthetic misspellings that represent these errors. The experiments conducted revealed a significant decrease in performance across all models for all types of noise, with F_1 drops ranging from 6.1 to 11.7, depending on the nature of the affected words. Additionally, their results indicated that the harm of synthetic misspellings is generally more pronounced than that caused by natural misspellings.

Satheesh *et al.* (2025) created a robustness benchmark for Question Answering based on misspellings of different types. The authors used BERT and other open-source models (Electra, Gelectra, Roberta-XLM), reporting a significant degradation in performance in all cases.

Both Liu *et al.* (2019) and Röttger *et al.* (2021) evaluate model performance on difficult data distributions, including misspellings. Liu *et al.* (2019) introduce a method called *inoculation by fine-tuning,* which involves creating normal and *challenging* versions of both training and test sets. The model is initially trained on the normal set and tested on both versions. If the performance is high on the normal set but low on the challenging set, the model is then fine-tuned using the challenging training set. This approach helps to determine whether the issue lies with the model itself or the original training data. If the model's performance improves with this fine-tuning, it suggests that data augmentation could be sufficient for the model to generalise better. The method was applied to NLI datasets (some proposed in Naik *et al.* (2018) and involved two models: the ESIM model of Chen *et al.* (2017b) and the decomposable attention model of Parikh *et al.* (2016). The results revealed that all the tested models struggle with synthetic misspellings, even if fine-tuned.

Röttger *et al.* (2021) tested the effectiveness of previously trained hate speech detection models when in the presence of misspellings. Specifically, the models are evaluated on 29 functional classes, including categories such as *hate expressed using denied positive statements* and *denouncements of hate that quote it*. This approach allows for very detailed results on the model's ability to detect different types of hate speech. Among the 29 classes, 5 are related to the presence of misspellings (called *spelling variations* in the paper). The models tested include BERT, Google's Perspective, and TwoHat's SiftNinja. The results revealed that all models struggle to handle misspellings, but the model Perspective fared significantly better than the others.

4.2 The harm of natural misspellings

Naturally occurring misspellings are invaluable resources for testing NLP applications in real-world settings (Baldwin *et al.* 2013; Belinkov and Bisk 2018). However, they are rarely employed in practice since collecting natural misspellings is anything but a simple task. With a lack of consensus on what precisely a *misspelling* is, some authors have considered as "natural misspellings" phenomena like the errors generated by second language learners (Náplava *et al.* 2021) or by OCR scans (Vinciarelli 2005). Having said this, natural misspellings may include all the types of misspellings listed in Section 7.4.2, as long as they are user-generated.

In this section, we review works that try to characterise the presence of natural misspellings (Section 4.2.1) and other studies that test the resiliency of different models to natural misspellings (Section 4.2.2).

4.2.1 Where do natural misspellings tend to occur?

Previous studies related to the analysis of natural misspellings are often devoted to understanding *which* types of misspellings are more likely to occur in which domains (Baldwin *et al.* 2013; Plank 2016).

Identifying where natural misspellings are most common is far from trivial, since the very notion of a *domain* is itself ambiguous. According to Plank (2016), real-world data emerge as complex interactions of many more dimensions (language, genre, register, age group, etc.) than what

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

we can realistically anticipate in an experimental setting. While certain domains of information, such as user-generated content, are known to be particularly prone to generating misspellings, the interplay of these dimensions means that *where* a misspelling occurs is often a matter of overlapping factors rather than a single one.

Baldwin *et al.* (2013) compared the rate of out-of-vocabulary terms (as a proxy of the number of misspellings) expected to be found in texts as a function of how curated these texts are. The results were arranged in an ordinal scale of increasing levels of curation: tweets, comments, forums, blogs, Wikipedia articles, and documents from the British National Corpus. In their study, the authors took into account some lexical units like the word length, the sentence length, and the rates of out-of-vocabulary terms, finding interesting direct correlations between the level of formality of the text and the average word and sentence length, with an anti-correlated rate of out-of-vocabulary terms. The same study analysed the perplexity of language models when processing different types of text. The results show that models trained and tested in similar domains (hence close to each other in terms of the degree of formality) tend to display lower perplexity. For example, a model trained on tweets (highly informal) shows a much lower perplexity when used to process blog forums (somewhat informal) than when used to process Wikipedia articles (highly formal).

4.2.2 Testing resilience against natural misspellings

Nguyen and Grieve (2020) studied how robust different word embedding techniques (such as word2vec variants and FastText) are to deviations from conventional spelling forms (including misspellings, among others) typical of social-media content. Using two datasets from Reddit and Twitter, the authors found that even techniques that are not specifically designed to take into account spelling variations (like the word2vec's skip-gram model) manage to capture them to some extent. Interestingly enough, the authors draw a connection between *intentional* spelling variations (like an *elongated* word *"goooood"*) and performance, suggesting that these variations typically arise in well-controlled situations, acting as a form of sentiment markers, and, for this reason, models are somehow able to make sense out of them. This is in contrast to unintentional misspellings, which are haphazardly distributed and tend thus to be harder to handle.

In addition to synthetic misspellings, Agarwal *et al.* (2007) carried out experiments on natural misspellings. To this aim, the authors used datasets from user-generated content, including logs from call centres, emails, and SMS. Their results suggest that real noise in user-generated content exhibits some patterns, attributable to the consistent usage of abbreviations and the repetition certain users make of specific errors. The results showed the models tested (SVM and Naive-Bayes) performed better than when confronted with synthetic misspellings (see Section 4.1).

In a similar vein, Ravichander *et al.* (2021) conducted experiments not only using synthetic misspellings (see Section 4.1) but also natural ones, in the context of question answering using the XQuAD dataset as a reference. The authors considered two types of natural misspellings: keyboard misspellings and automatic speech recognition (ASR) noise. For keyboard misspellings, natural misspellings were created by asking people to retype XQuAD questions without being able to correct their input when they made a mistake. For ASR, natural noise was created by reading and transcribing every question three times, by three different persons. The experiments showed a noticeable decrease in performance across all tested models (BiDAF, BiDAF-ELMo, BERT, and RoBERTa) for all types of noise. However, synthetic misspellings appeared, on average, to be slightly more problematic than natural ones. Among the types of natural noise, the one generated via ASR was found to be the most harmful. RoBERTa, the top-performing model of the lot, experienced a decay of 8 per cent terms in F_1 when confronted with such misspellings.

The study by Benamar *et al.* (2022) provides a detailed evaluation of how state-of-the-art subword tokenisers handle misspelt words. Specifically, they investigated two French versions of BERT (FlauBERT and CamemBERT) against natural misspellings originating in three different

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

domains (medical, legal, and emails). To test the tokenisation of misspellings, the authors randomly extracted 100 misspelt words from each corpus and paired them with their correct forms. The test was conducted by measuring the cosine similarity between tokens generated for the misspelt and clean terms. In all three domains, the average similarity was very low (19 per cent in the legal domain, 39 per cent in the medical domain, and 27 per cent in the email domain). The authors also observed that incorporating POS tagging information drastically helped to improve performance. For example, CamemBERT scored 92 per cent of the average cosine similarity in the email domain with the aid of POS tags.

4.3 The harm of hybrid misspellings

As recalled from Section 3.2, aside from the synthetic and natural misspellings, there is a third type of misspelling called *hybrid* that refers to real misspellings that have been artificially injected in different contexts for evaluation purposes (more on this in Section 7.4.3). To our knowledge, the only published record that employs hybrid misspellings to quantify the performance impact on systems that do not handle them is that of Belinkov and Bisk (2018). In their study, words from error correction databases (such as Wikipedia edits and second language learner corrections) were injected into the IWSLT 2016 machine translation dataset. While hybrid misspellings had less of an impact on machine translation tasks compared to synthetic misspellings, Belinkov and Bisk (2018) noted that hybrid and natural misspellings are more challenging to evaluate in a rigorous manner.

5. Methods

In this section, we offer a comprehensive overview of previous efforts devoted to counter misspellings. We organise existing methods according to the following categorisation:

- Data augmentation (Section 5.1): methods that enhance the training set with perturbed signals to develop resiliency to them. This group can be further divided into two subcategories:
  - Generalised data augmentation approaches (Section 5.1.1)
  - Adversarial training approaches (Section 5.1.2)
- Character-order-invariant representations (Section 5.2): methods devoted to counter one specific type of misspelling caused by variations in the natural order of the characters.
- Double step (Section 5.3): techniques that carry out a step of spelling correction (step 1) before solving the final task (step 2).
- Tuple methods (Section 5.4): methods that, in order to train a model, use a list of misspellings each annotated with the corrected surface form.
- Other methods (Section 5.5): relevant techniques that do not squarely belong to any of the above categories.

While, in principle, the methods are largely orthogonal to the tasks they have been applied to (more on this in Section 7.1), a few incidental patterns can be observed: for example, data augmentation methods have been more frequently tested in machine translation, whereas double-step methods have been applied across a wider variety of tasks. Methods specifically devoted to POS tagging are not homogeneous and are thus included in the "other methods" category. Table 1 provides a comprehensive overview of the associations between methodological principles, tasks, types of misspellings, models, datasets, and evaluation metrics, aggregating information from the papers discussed in this section, and is intended as a practical guide for the reader.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Table 1. Reference guide for the methods discussed in Section 5, along with tasks and misspellings addressed, type of models, datasets, and metrics used in the evaluation

Ref.	Section	Class	Task	Misspelling	Model	Datasets	Metrics
Heigold *et al.* (2018)	5.1.1	Data Augmentation	Machine translation	Swap, Middle-Perm.	Char-based CNN, BPT CNN	WMT16, newstest 2016	BLEU
Belinkov and Bisk (2018)	5.1.1	Data Augmentation	Machine Translation	Full-Perm., Swap, QWERTY	Char-based CNN	TED parallel corpus	BLEU
Vaibhav *et al.* (2019)	5.1.1	Data Augmentation	Machine Translation	Add., Del.	Back-translation LSTM	MTNT, TED, EP	BLEU
Karpukhin *et al.* (2019)	5.1.1	Data Augmentation	Machine Translation	Swap, Del.	Char-based CNN	IWSLT	BLEU
Li and Specia (2019)	5.1.1	Data Augmentation	Machine Translation	Various	Transformer	MTNT	BLEU
Zheng *et al.* (2019)	5.1.1	Data Augmentation	Machine Translation	Various	Transformer	WMT15, KFTT, TED parallel corpus, JESC, MTNT	BLEU
Namysl *et al.* (2020)	5.1.1	Data Augmentation	Named Entity Recognition Sequence Labelling	Full-Perm., Middle-Perm., Swap, QWERTY	BiLSTM-CRF BERT, FLAIR, ELMo, Glove	CoNLL 2003, 2004, German Eval 2004	F1
Passban *et al.* (2021)	5.1.1	Data Augmentation	Machine Translation	Add., Del.	BPE + Transformer	WMT-14, newstest 2013, 2014	BLEU
Cheng *et al.* (2019)	5.1.2	Adversarial Training	Machine Translation	Various	Transformer	LDC, NIST 2002-2008, WMT14, newstest 2013, 2014	BLEU
Li *et al.* (2019a)	5.1.2	Adversarial Training	Text Classification	Adversarial	LR CNN LSTM	Rot. Tom. Mov. Rev. IMDB	Success Rate Accuracy
Park *et al.* (2020)	5.1.2	Adversarial Training	Machine Translation	Del., Add., Swap	Transformer	MTNT 2018, 2019, IWSLT 2013, 2015, 2017	BLEU
Zhou *et al.* (2020)	5.1.2	Adversarial Training	Named Entity Recognition	Natural Misspellings	CNN LSTM HN	CoNLL 2002, 2003, WNUT 2016, 2017	F1

Table 1. Continued

Ref.	Section	Class	Task	Misspelling	Model	Datasets	Metrics
Sakaguchi *et al.* (2017)	5.2	Char-order agnostic	Grammatical Error Correction	Middle-Perm., Del., Add.	Char-based, CNN, ScRNN	PT	Accuracy
Belinkov and Bisk (2018)	5.2	Char-order agnostic	Machine Translation	Full-Perm., Middle-Perm., Swap, QWERTY	Char-based CNN	IWSLT 2016	BLEU
Malykh *et al.* (2018)	5.2	Char-order agnostic	Sentiment Analysis, Text Entailment	Natural Misspellings	LSTM, SRU, CNN	Reuters RCV1, Russian National Corpus, Turkish "42 bin haber"	ROC AUC
Sperduti *et al.* (2021)	5.2	Char-order agnostic	Various	Full-Perm., Middle-Perm.	Word2vec	British National Corpus	Various
Schulz *et al.* (2016)	5.3	Double-step	POS Tagging, NER, Lemmatisation	Various	Char. based models	SMS corpus, Twitter Corpus, ask.fm dataset	Accuracy, Recall
Mizumoto and Nagata (2017)	5.3	Double-step	Machine Translation	Various	Various	Konan-JIEM (KJ) corpus, propietary	BLEU
Ljubesic *et al.* (2017)	5.3	Double-step	POS tagging, Morphosyntactic descriptions	Various	Char. based models	ssj500k, Janes-Tag	Accuracy, Recall
Pruthi *et al.* (2019)	5.3	Double-step	Sentiment Analysis, Paraphrase Detection	Swap, Add., Del., QWERTY	BERT, Bi-LSTM	Stanf. Sent. Tree., MRPC	Accuracy
Kurita *et al.* (2019)	5.3	Double-step	Text Classification	Full-Perm.	BERT, ELMO, FastText, CDAE	Jigsaw 2018, 2019, OffensEval 2019	Accuracy
Riordan *et al.* (2019)	5.3	Double-step	Automated Content Scoring	Various	Neural Scoring Models	Automated Student Assessment Prize	Mean Squared Error
Riordan *et al.* (2019)	5.3	Double-step	Automated Content Scoring	Various	Neural Scoring Models	Automated Student Assessment Prize	Mean Squared Error
van der Goot *et al.* (2020)	5.3	Double-step	Dependency Parsing, Lexical normalisation	Various	Lexical Normalisation Model, Standard dependency parser	NormIT!	Mean Squared Error

Table 1. Continued

Ref.	Section	Class	Task	Misspelling	Model	Datasets	Metrics
Li *et al.* (2021)	5.3	Double-step	Machine Translation	Various	Transformer, GRU	flickr2017, mscoco2017	BLEU, METEOR
Passban *et al.* (2021)	5.3	Double-step	Machine Translation	Add., Del.	BPE + Transformer	WMT-14, newstest 2013, 2014	BLEU
Mamta *et al.* (2023)	5.5	Double-step	Sentiment Classification, Hate speech detection	Various	BERT, RoBERTa	Code-mixed datasets (Hinglish; Benglish)	Accuracy
Zhou *et al.* (2019)	5.4	Tuple methods	Machine Translation	Various	Transformer	TED, MTNT	BLEU
Edizel *et al.* (2019)	5.4	Tuple methods	Word Similarity	Other	FastText, MOE	Wikipedia, WS353, Rare Words, Mikolov's word analogy	Various
Doval *et al.* (2020)	5.4	Tuple methods	Various	Various	Word2vec, FastText	wordsim353, SCWS, SimLex999, SemEval17	Spear.
Alam and Anastasopoulos (2020)	5.4	Tuple methods	Machine Translation	Various	Bart For Conditional Generation	NUCLE, FCE, Lang8, JFFLEG-ES	Various
Li *et al.* (2016)	5.5	Other methods	Various	Other	CNN	Subj, CR, Stanf. Sent. Tree., Rot. Tom. Mov. Rev.	Accuracy
Jones *et al.* (2020)	5.5	Other methods	Various	Swap, Add., Del.	BERT	GLUE	Various
Wang *et al.* (2020)	5.5	Other methods	Machine Translation	Natural Misspellings (various types)	CNN-LSTM	IWSLT 2016	BLEU
Sankar *et al.* (2021)	5.5	Other methods	Text Classification, Data Augmentation	Swap, Add., Del.	BERT, BiLSTM, SGNN, ProSeqo	MRDA, SWDA, AR, YA	Accuracy

Table 1. Continued

Ref.	Section	Class	Task	Misspelling	Model	Datasets	Metrics
Salesky *et al.* (2021)	5.5	Other methods	Machine Translation	Swap, Full-Perm., Middle-Perm.	fairseq	MTTT TED, MTNT, WIPO, WMT	BLEU
Riabi *et al.* (2021)	5.5	Other methods	POS Tagging	Various	CharacterBERT, CamemBERT, mBERT	Nazrabi Treebank	Accuracy
Sidiropoulos and Kanoulas (2022)	5.5	Other methods	Question Answering	Full-Perm., Middle-Perm., QWERTY	BERT	MS MARCO, Natural Questions	MMR, Recall, Top k-ranks
Chen *et al.* (2022)	5.5	Other methods	Sentiment Analysis, Text Classification, NER, Word Similarity, Word Cluster	Swap, Del., Add., QWERTY	FastText, MIMIC, BoS, KVQ-FH, LOVE	Stanf. Sent. Tree. 2, Rot. Tom. Mov. Rev., CoNNL 2003, BC2GM, Various Intrinsic Datasets	Accuracy, F1, Spearman, Purity
Aepli and Sennrich 2022)	5.5	Other methods	POS Tagging	Various	DE-BERT, mBERT	MOROCO, SwissCrawl, The Credit Suisse News Corp., NOAH's Corpus	Accuracy, F1, Spearman, Purity
Bernhard and Dolińska (2025)	5.5	Other methods	POS Tagging	Various	XLM-RoBERTa	Alsatian corp., Dagur corp.	Accuracy
Muñoz-Ortiz et al., (2025)	5.5	Other methods	POS tagging, Dependency Parsing, Topic Classification, Intent detection	Various	PIXEL	DBMDZ German corpus, GSD and HDT	Accuracy
Pagnoni *et al.* (2025)	5.5	Other methods	Commonsense Natural Language Inference	Various	LLama3 Llama3.1 BLT-Mono	HellaSwag	Accuracy

5.1 Data augmentation

One of the earliest attempts to address the problem of misspellings in NLP comes down to expanding the training set with misspelt instances, so that the model learns to deal with them during training.

Although data augmentation techniques typically lead to direct improvements, there are important limitations worth mentioning. Augmenting the training set entails an additional cost, sometimes derived from complex techniques that seek to uncover the weaknesses of the model. Yet another important limitation regards its circumscription to a limited frame time. The misspelling phenomenon is not stationary, since language is in constant evolution. While core spelling conventions in languages like English remain relatively stable over long periods, vocabulary changes, slang, dialectal variations, and even official spelling reforms in some languages introduce orthographic shifts that impact misspellings and their treatment in NLP (for a more detailed discussion of these diachronic aspects, see Sections 3.1.2 and 6.5). Additionally, data augmentation typically over-represents certain types of misspellings, thus injecting *sampling selection bias* into the model (i.e., the prevalence of the phenomena represented in the training set widely differs with respect to the prevalence expected for the test data as a result of a selection policy). Finally, misspellings consist of different character combinations, making it nearly impossible to achieve comprehensive coverage.

We first review a direct application of data augmentation strategies to the problem of misspellings (Section 5.1.1) and then move to describing methods that use a specific kind of generation procedure based on adversarial training (Section 5.1.2)

5.1.1 Generalised data augmentation

To the best of our knowledge, the first attempt to cope with misspellings by means of data augmentation is by Heigold *et al.* (2018). The methodology consists of analysing the type of misspellings that most harmed the performance of a machine translator, and inserting similar occurrences in the training set. In a similar vein, Belinkov and Bisk (2018) injected misspellings of various types in a parallel corpus, including the *full permutation*, *character swapping*, *middle permutation*, and *insertion of QWERTY errors*. Information about how precisely these misspellings are individuated, and about other types of misspellings, is available in Section 7.4 devoted to datasets.

Data augmentation has been extensively applied to the problem of machine translation (Karpukhin *et al.* 2019; Li and Specia 2019; Passban *et al.* 2021; Vaibhav *et al.* 2019; Zheng *et al.* 2019) as a means to confer resiliency to misspellings to the models (for the most part, Character-based neural approaches). For example, Vaibhav *et al.* (2019) augment the training instances of French and English languages in the EP dataset (see Section 7.4) by using the MTNT dataset of Michel and Neubig (2018) (see Section 7.4). Karpukhin *et al.* (2019) experimented with four different types of misspellings, correspondingly generated by *deleting*, *inserting*, *substituting*, and *swapping* characters, that were applied to 40 per cent of the training instances for Czech, German, and French source languages. Some authors have investigated the idea of *backtranslation* (i.e., reversing the natural direction of the translation, thus translating from the target language to the source language) as a mechanism to generate additional data. The idea is to generate the source translation-equivalent in domains in which resources for the target language are more abundant. The final goal is thus to enhance the source data and to inject misspellings so that a machine translation model resilient to misspellings can be later trained (Li and Specia 2019; Zheng *et al.* 2019). In particular, Zheng *et al.* (2019) applied this technique to social media content for English-to-French, based on the observation that training data for this social media rarely contain misspellings in the target side, or do so in very limited quantities. They used additional techniques to expand the training set, including the use of out-of-domain documents (they considered the domain of news) along with their automatic translations.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Similarly, Li and Specia (2019) combined the idea of backtranslation with a method called *Fuzzy Matches* (Bulté and Tezcan 2019). Fuzzy Matches takes as input a parallel corpus and a monolingual dataset and, for each sentence in the monolingual dataset, searches for the most similar ones in the parallel corpus, and returns the translation equivalent (i.e., its parallel view) as a potential translation for the original sentence. This method was applied to a monolingual corpus containing misspellings either *backwards* (this happens when the monolingual corpus is from the target language) and *forward* (this happens when the monolingual corpus is from the source language), thus generating (clean) translation approximations of noisy data. They combined this heuristic with a method to generate a monolingual corpus based on generating automatic transcriptions from audio files (using the so-called Automatic Speech Recognition software), in the hope that these transcriptions would eventually contain misspellings.

A different approach for developing resiliency to misspellings is the so-called *fine-tuning* approach that, in the context of machine translation, comes down to using a pre-trained translator model (typically trained on clean data) and performing additional epochs of training using source instances with injected misspellings (Namysl *et al.* 2020). Passban *et al.* (2021) experimented with a variant of this approach, called *Target Augmented Fine-Tuning* (TAFT), that consists of concatenating, at the end of the target sentence, the correct spelling of the misspelt term of the source sentence. The idea is to condition the model not only to produce the target sentence but also to discover the correct spelling of the affected source word.

Data augmentation has been applied to problems other than machine translation as well. For example, Namysl *et al.* (2020) propose a mechanism for generating misspelt entries for the tasks of named entity recognition (NER) and neural sequence labelling (NSL) characters of the words in a sentence as follows. Given a word $w = (c_1, \ldots, c_n)$ consisting of n characters, a pseudo-character ϵ is inserted before every character and after the last one, thus obtaining a new token $w = (\epsilon, c_1, \epsilon, c_2, \epsilon, \ldots, c_n, \epsilon)$. For example, given the word *spell*, a token $\epsilon\ s\ \epsilon\ p\ \epsilon\ e\ \epsilon\ l\ \epsilon\ l\ \epsilon$ is created. Subsequently, a few of these characters are randomly chosen and replaced with another character randomly drawn from a certain probability distribution (called the *character confusion matrix*) that also includes ϵ in its domain. For example, two possible derivations would be (note the underlined characters):

$$\text{(i)}\ \epsilon\ s\ \underline{\epsilon}\ p\ \epsilon\ e\ \epsilon\ l\ \epsilon\ l\ \epsilon\ \rightarrow\ \epsilon\ s\ \underline{m}\ p\ \epsilon\ e\ \epsilon\ l\ \epsilon\ l\ \epsilon$$
$$\text{(ii)}\ \epsilon\ s\ \epsilon\ p\ \epsilon\ e\ \epsilon\ l\ \epsilon\ \underline{l}\ \epsilon\ \rightarrow\ \epsilon\ s\ \epsilon\ p\ \epsilon\ e\ \epsilon\ l\ \epsilon\ \underline{\epsilon}\ \epsilon$$

Finally, all the remaining pseudo-characters are removed. In our example, this would give rise to the words (i) *smpell* and (ii) *spel*, respectively.

5.1.2 Adversarial training

A different, related strategy for augmenting the data is by means of *adversarial training*. Adversarial training is a robustness-oriented learning paradigm in which a model is trained not only on the original (clean) data, but also on adversarially perturbed variants of it. These perturbations are deliberately crafted to exploit the model's weaknesses and induce errors, with the goal of improving its resilience. In the context of misspellings, adversarial training typically involves injecting orthographic perturbations into the training data so that the model learns to maintain performance despite such input variations. There are two main types of adversarial training that have been applied to the problem of misspellings: the black-box setting and the white-box setting, which we discuss in what follows.

In **the black-box setting**, perturbations are created without direct access to the model, often by applying predefined transformation rules or by using surrogate models. Here, a general-purpose model is trained to develop robustness to adversarial samples.

Li *et al.* (2019a) propose TextBugger, a method to generate misspellings by means of adversarial attacks. The method first searches for the most influential sentences (those for which the classifier

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

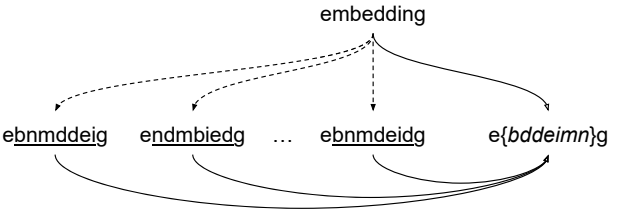

Figure 2. Conceptualisation of an order-agnostic representation for garbled words. Dotted lines denote garbled variants of the original word, on the top. Solid lines denote an order-agnostic representation of a surface form word. If all characters (but the first and last) are represented as a set, then the representation of the original word and the garbled variants coincide.

returns the highest confidence scores) and then identifies the most important words in each such sentence (those that, if removed, would lead to a change in the classifier output). These words are altered by injecting misspellings either in training or in test documents.

In the **White-box setting**, perturbations are generated using knowledge of the model's parameters or gradients, and the objective function is a *perturbation-aware loss*, that is, a loss that jointly optimises performance on clean inputs and on their adversarially perturbed counterparts. This contrasts with the traditional loss, which only accounts for clean inputs.

Zhou *et al.* (2020) generate adversarial examples via a perturbation-aware loss following Goodfellow *et al.* (2015), that is, a perturbation to the input *optimised for damaging the loss* of the model. Their neural model was dubbed *Robust Sequence Labelling* (RoSeq) and was applied to the problem of named-entity recognition (NER). The idea is to optimise both for the original model's loss and for the perturbation loss, simultaneously. Note that in this case, there is no explicit augmentation of training data, but rather an implicit regularisation in the loss function that carries out the adversarial training approach.

Cheng *et al.* (2019) applied a similar idea but in the context of machine translation. The method is called *Doubly Adversarial Input* since, in this case, the perturbation is applied both to the source and to the target sentences. The most influential words in a sentence (hence, the candidates to perturb) are identified by searching for possible replacements that, if used in place of the original word, would yield the maximum (cosine) distance in the embedding space with respect to the original vector. The set of candidate words that are electable for this replacement is made of words that are likely to occur in place of the original one according to a language model trained for the source or target language, correspondingly. For the target sentences, this set is further expanded with words that the translator model itself considers likely. Later on, Park *et al.* (2020) extended this idea to the concept of *subwords* and their segmentation (see also Kudo and Richardson 2018).

5.2 Character-order-agnostic methods

There is abundant evidence from the psycho-linguistics literature indicating that humans are able to read garbled text (i.e., text in which the character-order within words is rearranged, often called *scrambled*) without major difficulties, as long as the first and last letters remain in place, as in, *The chatrecras in tihs sencetne hvae been regarraned.* (see, e.g., Andrews 1996; McCusker *et al.* 1981; Rayner *et al.* 2006). This is not true for computational language models relying on current representation mechanisms, though (Heigold *et al.* 2018; Yang and Gao 2019). Character-order-agnostic methods (Belinkov and Bisk 2018; Malykh *et al.* 2018; Sakaguchi *et al.* 2017; Sperduti *et al.* 2021) gain inspiration from these observations and propose different mechanisms that defy the need for representing the internal order of the characters; Figure 2 depicts this intuition.

The earliest published work we are aware of is by Sakaguchi *et al.* (2017). Their model, called the *Semi-Character Recurrent Neural Network* (ScRNN), represents the first and last characters of a word as separate one-hot vectors, while the internal characters are encoded as a bag-of-characters, that is, a juxtaposition of one-hot vectors where character order is disregarded. The model was

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

applied specifically to spelling correction, rather than to any particular downstream application. ScRNN was adopted as the first stage of a double-step method by Pruthi *et al.* (2019) (covered in Section 5.3).

Later on, Belinkov and Bisk (2018) proposed a representation mechanism, called *meanChar*, that was tested in machine translation contexts. In particular, the representation comes down to averaging the character embeddings of a word, and then using a word-level encoder, along the lines of the CharCNN proposed by Kim (2014).

Malykh *et al.* (2018) proposed *Robust Word Vectors* (RoVe), a method that generates three vector representations out of each word: the *Begin* (B), *Middle* (M), and *End* (E) vectors. These vectors correspond to the juxtaposition of the one-hot vectors of certain characters in a word. For example, given the word *previous*, B is the sum of the one-hot vectors of the first three characters (*pre*), E is the sum of the one-hot vectors of the last three characters (*ous*), while M sums the one-hot vectors of all characters in the word (and not only of the remaining central characters, as the name may suggest). The method showed promising results in three different languages, including Russian, English, and Turkish, and in three different tasks, including paraphrase detection, sentiment analysis, and identification of textual entailment.

Sperduti *et al.* (2021) proposed a pre-processing trick, called *BE-sort*, to tackle the problem. The method comes down to alphabetically sorting all middle characters of a word, excluding the first and the last character, so that the original word itself (e.g., *embedding*) as well as any potentially garbled variant of it (e.g., *edbemindg*, *ebmeinddg*, etc.) would end up being represented by the exact same surface token (e.g., *ebddeimng*). This pre-processing is not only applied to the words in the training corpus, but also to every future test word. Word embeddings learned by using Skip-gram with negative sampling on a BE-sorted variant of the British National Corpus were found to perform almost on par, across 17 standard intrinsic tasks, with respect to word embeddings learned on the original corpus, and much better than word embeddings learned on variants of this corpus in which words were garbled at different probabilities.

5.3 Double-step with text normalisation

As the name suggests, double-step methods tackle any task by performing two subsequent steps: first, a task-agnostic *text normalisation* step addresses and corrects any misspellings in the input text; second, the actual task of interest is performed, with the assumption that the input is now error-free. Since the first step removes misspellings from the source text, some authors have suggested that double-step methods represent the opposite of data-augmentation-based approaches. For example, Plank 2016 analyses the problem by which models are trained on clean (canonical) data, but tested on potentially noisy data, and suggests that a key component for enabling resiliency to out-of-vocabulary terms and adaptation to language variation would come down to modelling variety (i.e., enlarging the training data) rather than simply cleaning the test.

In this survey, we cover spelling correction methods only when they are specifically aimed at improving the performance of a downstream task (i.e., when they serve as the "first step" in a double-step approach), and we refer readers interested in general-purpose correction methods to Bryant *et al.* (2023); Hládek *et al.* (2020); Wang *et al.* (2021b). This in no way diminishes the importance of spelling correctors; indeed, in most application contexts, spelling correction directly improves final task performance and is often sufficient for many industrial solutions (Bhargava *et al.* 2017). However, our survey focuses on the implications of misspellings for the entire processing pipeline, that is, cases where removing misspellings might lead to the loss of potentially useful signals for the target task. Some examples of relevant applications are provided in Section 9.

There are two main strategies for implementing double-step methods. The first one, which we could call the *independent* approach, in which the error correction step is carried out independently from the second, task-specific step, which receives the cleaned input. The second one,

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

which we call the *end-to-end* approach, instead considers the correction step and the task-specific step as dependent, and optimises both jointly. In most cases, the methods have used the first strategy; for example, Schulz *et al.* (2016) proposed a new modular text correction method to serve as the *first step* in a double-step process. The correction method is structured into three *internal layers*: (i) a preprocessing layer, which performs text tokenisation; (ii) a suggestion layer, which generates several possible corrections; and (iii) a final decision layer, which selects the best correction. After this correction stage, the *second step* of the double-step process consists of training and testing POS tagging and NER models on the normalised text. The authors showed that their modular correction method improves robustness to misspellings in the downstream task.

Ljubesic *et al.* (2017) evaluated a standard tagger on non-standard Slovene text and observed a clear drop in accuracy. To address this, the authors incorporated lexical normalisation data, aligning non-standard word forms with their standardised counterparts, either through lexicon-based mappings or automatically generated normalisations. Adding this information to the tagger's feature set improved its ability to handle spelling variants and colloquial forms, leading to notable gains in POS tagging accuracy.

Later on, Riordan *et al.* (2019) set up an experiment inserting Character-based representations into neural word-based content scoring models,[8] evaluating whether text correction alone or in combination with character-level modelling provides greater improvements on responses that include misspellings. While Character-based information appears to have minimal impact, spelling correction improves the models' resilience to misspellings in the downstream task.

Pruthi *et al.* (2019) adopted a variant of ScRNN (covered in Section 5.2) as the first step of a double-step strategy applied to the problem of sentiment analysis and part-of-speech tagging. The variant implements heuristics for handling the *unknown tokens* (typically denoted by UNK) that ScRNN produces whenever it encounters an out-of-vocabulary (OOV) word (i.e., words that were not considered during the training phase). In particular, three mechanisms are explored: (i) *pass-through*, in which the UNK token is replaced by the original OOV term; (ii) *back-off to neutral*, in which the UNK token is replaced by a word that has a neutral value for the classification task; and (iii) *back-off to a background model*, in which another, more generic (hence less suitable for the task), spelling corrector is invoked in place of ScRNN.

Kurita *et al.* (2019) propose the *Contextual Denoising Autoencoder*. The Autoencoder receives as input the incorrect version of a textual token (e.g., *wrod*, incorrect spelling of a *word*) and predicts its denoised version in the output (e.g., *word*) by leveraging contextual information. The base architecture that Kurita *et al.* (2019) employed is a transformer model. To embed words, Kurita *et al.* (2019) exploited the CNN encoder of ELMO. van der Goot *et al.* (2020) created a new lexical normalisation benchmark for the Italian language and showed how a normalisation step can slightly improve resiliency to misspellings in dependency parsing. Both Li *et al.* (2021) and Passban *et al.* (2021) propose methods for machine translation that take into account error correction in an *end-to-end* manner. Both approaches resort to an auxiliary task based on a double decoder for correcting the input. Given the noisy instance x', the decoder is trained to produce its translated version y, while the correction decoder is trained to regenerate x, the clean version of x'. The two decoders are jointly optimised by means of a weighted loss that takes into account the translation error and the reconstruction loss simultaneously.

5.4 The tuple-based methods

By *tuple-based methods*, we refer to a broad family of approaches in which the input data is represented as tuples that explicitly list relevant spelling variations. This term does not characterise a specific learning paradigm but rather describes a representational format for the data; as such, it places no constraint on the type of method used to learn from these data. Consequently,

[8] Content scoring is the task of automatically evaluating human-generated text, such as college essays or responses to academic test questions.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

the methods grouped under this section are diverse in nature. The two most common formats for representing training data are: (i) *pairs* of the form (x, x'), where x is a clean instance and x' is a misspelt variant, and (ii) *triplets* of the form (x, x', y) where y is a task-dependent target (e.g., a translation of x). Here, x and x' can be words, sentences, or other lexical units.

Alam and Anastasopoulos (2020) used a tuple-based method to endow a transformer-based machine translator with resiliency to misspellings. To do so, they resorted to a dataset originally designed for grammatical error correction and consisting of tuples (x, x'), with x' a misspelt version of the sentence x. The idea is to generate translations of x to create new tuples (x', y) in which the misspelling-free translation y is presented as the desired output for the misspelt input x'; tuples thus created are then used to fine-tune a transformer model.

Zhou *et al.* (2019) proposed a cascade model based on triples for machine translation. Given a triplet (x, x', y) (in which x, x', and y are defined as before), the model combines two auto-encoders sequentially: the first one is a denoising auto-encoder that receives x as the expected output for input x', while the second one is a translation decoder that receives y as the expected output for the encoded representations of x and x'.

Edizel *et al.* (2019) propose *Misspelling Oblivious word Embeddings* (MOE), a variant of FastText (Bojanowski *et al.* 2017; Joulin *et al.* 2017), which, in turn, is a variant of the CBOW architecture of word2vec (Mikolov *et al.* 2013) that endows the architecture with the ability to model subword information. The idea is to enhance the loss function of FastText with a component that favours the embeddings of subwords from misspelt terms to be close to the embedding of the correct term. To this aim, the authors created a dataset of word tuples (x, x') by relying on a probabilistic error model that captures the probability of mistakenly typing a character c' when the intended character was c by taking into account the entire word and its context. The probabilistic model was developed using an internal query log of Facebook.

Closely related, Doval *et al.* (2020) propose a modification of the Skip-Gram with Negative Sampling (SGNS) architecture of word2vec (Mikolov *et al.* 2013) based on triplets of the form (w, w', b_j), where b_j is the j-th word in a set of *bridge words*, and w' is a misspelt version of word w. The intuition behind bridge words is as follows. Consider the occurrence of the word *friend* in a document that also contains the misspelt form *frèinnd* in similar contexts; consider, for example, the sentence *my friend is tall* and *my frèinnd is tall*. The method first pre-processes the text by eliminating double letters and accents. In our example, *frèinnd* would thus become *freind* (note that two letters remain swapped). Then, two sets of *bridge words* are generated, each containing all the words that would result from eliminating one single character from *friend* and *freind*, respectively. In our example, this would lead to one set of bridge words for the clean word *friend*, that is, {*riend*, *fiend*, *frend*, *frind*, *fried*, *frien*}, and another one for the misspelt word *freind*, that is, {*reind*, *feind*, *frind*, *freid*, *frein*}. All the words included in the union of both bridge word sets are then given as input to the SGNS model, which is requested to predict the target context (*my*, *is*, *tall*). The name *bridge words* refers to the fact that there are common elements at the intersection of both sets of variants, thus created (e.g., {*frind*}) which act as *bridges* between the correct and the misspelt variant. Some limitations of this method include the possibility to generate bridge words that collide with other existing words (e.g., *fiend*), and the increased computational cost that derives from the generation of potentially many new training instances. To counter these problems, the authors propose some heuristics, like generating bridge words only for a limited number of terms, and limiting the impact of the bridge words during training.

5.5 Other methods

This section is devoted to discussing relevant methods that do not belong to any of the aforementioned groups. Papers in this section include ideas as variegate as experimental encodings (Jones *et al.* 2020; Sankar *et al.* 2021; Salesky *et al.* 2021; Wang *et al.* 2020), regularisation functions (Li *et al.* 2016), and contrastive learning (Chen *et al.* 2022; Sidiropoulos and Kanoulas 2022).

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Jones *et al.* (2020) propose Robust Encoding (*RobEn*), a (context-free) encoding technique that maps a word (e.g., *bird*) along with its possible misspellings (e.g., *brid*, *bidr*, etc.) to the same token, so that the variability among these surface forms becomes indistinguishable to the downstream model. Unique tokens therefore represent *clusters* of terms and typos. The authors study means for obtaining these clusters, and analyse the impact of different clustering strategies in terms of *stability* (measures the resiliency to perturbations) and *fidelity* (a proxy of the quality of the tokens in terms of the expected performance in downstream tasks). An initial solution is proposed in which clusters are decided by seeking connected nodes in a graph in which nodes represent words from a controlled vocabulary, and in which edges connect words that share a common typo. Such a solution is found to lead to very stable solutions, but at the expense of fidelity. The final proposed method relies on agglomerative clustering and searches for the clusters by optimising a function that trades off stability for fidelity. Sankar *et al.* (2021) propose a method based on Locality-Sensitive Hashing (LSH). The final goal of LSH representations is to derive vectorial representations *on-the-fly*, thus reducing the memory footprint that traditional embedding matrices require. In a nutshell, LSH assigns a hash code (i.e., binary representation much shorter than a standard one-hot encoding) to a word based on its n-grams, skip-grams, and POS tags, and derives a vector representation as a linear combination of learnable (low-dimensional) basis vectors. The intuition is that LSH projections may lead to similar representations for clean and misspelt sentences, since this hashing is, by construction, low sensitive to noise. The experiments reported in text classification and perturbation studies seem indeed to confirm these intuitions.

Wang *et al.* (2020) experimented with visually-grounded embeddings of characters. The idea consists of generating an image for every character (i.e., rastering a character in a specific font type and font size), thus obtaining a matrix representation of it (the pixels of the image), that can directly be used as an embedded representation of the character. The intuition is that visually similar characters (e.g., "o", "O", "0") should end up being represented by similar such embeddings. The images (i.e., the character embeddings) are further reduced using PCA, and given as input to a Character-based CNN that acts as the encoder for a machine translation neural model. In a related study, Salesky *et al.* (2021) explored visually-grounded representations of sliding windows (specifically, subword tokens) for machine translation. This work was later used by the same team of researchers as the basis of PIXEL (Rust *et al.* 2023), a language model that similarly processes text as a visual modality. PIXEL was trained using the ViT-MAE (He *et al.* 2022) architecture on the same dataset used to train BERT, and demonstrated strong resilience to graphical misspellings, that is, cases where characters are visually similar. Muñoz-Ortiz *et al.* (2025) used the PIXEL models (described in Section 5.5) to see if they had an effect on non-standard text, showing that PIXEL is promising for dealing with non-standard text (including misspellings) in zero-shot contexts for the German language and several German dialects.

Li *et al.* (2016) investigate a special-purpose regularisation of the loss function that aims to confer resiliency to the presence of misspellings. The regularisation terms gain inspiration from adversarial training in computer vision, and are based on minimising the Frobenius norm of the Jacobian matrix of partial derivatives of the outputs with respect to the (perturbed) inputs. Although the method was found to work better than other regularisation techniques (including dropout), the method was only tested against *masking* misspellings, that is, against one specific type of noise consisting of replacing random characters with a mask symbol. It thus remains to be seen the extent to which this regularisation technique is of help when confronted with more general types of misspellings (e.g., swapping, garbling, deletion, etc.).

Sidiropoulos and Kanoulas (2022) studied ways for improving the performance of passage retrieval when the user questions contain misspellings. The approach is based on a combination of data augmentation and contrastive learning in dual-encoder architectures. The dual-encoder

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

is based on BERT and is trained to rank, given a user question, the correct passages higher than the incorrect passages. The data augmentation strategy consists of randomly deciding when to issue a clean question, or instead a misspelt variant of it, to the dual-encoder during training. The contrastive learning enforces the original question to be closer to the typoed variant than to any other question in the dataset. The experimental results prove that both data augmentation and contrastive learning help to improve the performance of passage retrieval, and that these techniques work even better when combined. In a related paper, Chen *et al.* (2022) propose LOVE, a contrastive method for learning out-of-vocabulary embeddings for misspellings that enforces representations of typoed words to be close to the representations BERT derives for the corresponding correct word.

A related body of papers has to do with the treatment of misspellings in the context of spam detection. Most of these papers belong to the first era of misspellings (see Section 2.1). For example, Ahmed and Mithun (2004) and Renuka and Hamsapriya (2010) rely on word stemming as a method to improve spam message detection, while Lee and Ng (2005) use a Hidden Markov Model-based method to correct misspelt words before performing detection.

Recent survey papers like those by Crawford *et al.* (2015) and Wu *et al.* (2018) indicate that analysing the textual content of a spam message is only one small part of the operations typically used for spam detection. In addition to text, other elements such as key segments (e.g., URLs), patterns in usernames, account statistics, etc., are also worth considering.

More recently, Mamta *et al.* (2023) introduced a technique for incorporating auditory features into language models to improve performance on code-mixed text (text produced in different sociolinguistic contexts, in this case blending Hindi or Bangla elements with English). Their approach involves training a BERT architecture with phonetic features extracted using the SOUNDEX algorithm. The resulting models demonstrated resilience to code-mixed data across various tasks and datasets.

Pagnoni *et al.* (2025) proposed a new type of tokeniser for LLMs, the so-called *Byte Latent Transformer* (BLT). BLT has no fixed vocabulary, but dynamically segments text into byte-based patches using entropy estimates. Their approach allows models with only 1B tokens to perform better or comparably to models with up to 16B tokens (LLaMA3.1) when confronted against different types of misspellings on the HellaSwag dataset (Zellers *et al.* 2019).

6. Misspelling, *error de ortografía*, *salah eja*: the multilingual problem

As in most branches of NLP, the vast majority of methods and evaluations assessing the impact of misspellings have, to date, focused primarily on English. This section turns to the topic of multilinguality in the context of misspellings, with two main goals: (i) to identify *which* additional languages are covered by the papers discussed so far, and to compare their coverage to that of English; and (ii) to present studies that take an explicitly multilingual-aware approach.

We dedicate separate sections to works that specifically address multilinguality (Section 6.1), cross-linguality for low-resource languages (Section 6.2), as well as L1 learner errors in the context of downstream tasks impacted by misspellings (Sections 6.3 and 6.4). We also include a section on spelling reforms (Section 6.5), which, although not traditionally considered misspellings, can introduce orthographic variation with significant implications for NLP systems.

Note that multilinguality is an orthogonal dimension with respect to other aspects discussed throughout the paper; therefore, this division is not incompatible with methodological aspects, applications, or evaluation strategies covered in other sections. That said, this section aims to raise awareness of the multilingual-related challenges involved in handling misspellings, rather than to provide an exhaustive survey of misspelling studies within the context of multilinguality.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

6.1 Quantifying multilingual coverage in misspelling research

In order to provide a rough estimate of language coverage in the field, we rely on the 42 papers surveyed in this work as a sample that we hope is reasonably representative, while still acknowledging that it may not fully reflect the actual distribution.

As discussed in Section 7, many of the reviewed studies on robustness to misspellings are machine translation studies, which inherently involve multiple languages. However, outside the context of machine translation, multilingual representation remains limited, with non-Western languages being particularly underrepresented.

Among the 42 papers summarised in our methods section, 28 include languages other than English. Nevertheless, only three of these (Malykh *et al.* 2018; Namysl *et al.* 2020; Zhou *et al.* 2020) are not focused on machine translation. The most frequently studied languages beyond English are German (12) and French (11), followed by Czech (5). Other represented languages include Arabic (2), Turkish (2), Spanish (2), Italian (2) and one instance each of Vietnamese, Polish, Dutch, Dagur, Alsatian, Nazrabi, Moldovan, Romanian, Hinglish, Benglish, Japanese, Slovenian and Portuguese.

These data underscore a notable imbalance: while some work does address languages other than English, the vast majority of the world's languages remain largely underrepresented in misspelling-related research.

6.2 Low-resource languages and cross-lingual approaches

Given that there are more than 7,000 living languages in the world, and that modern approaches—especially neural ones—are both computationally expensive and data-intensive, devising effective solutions for low-resource languages can appear daunting. In this context, cross-lingual approaches (i.e., methods that transfer knowledge from high-resource languages to low-resource ones) represent a particularly promising way out of this problem, if not the only viable one. Most cross-lingual approaches consider the *source* (typically a resource-rich language on which training is performed) and the *target* (typically a resource-scarce language on which the model is to be deployed) to belong to the same language family.

One of the cross-lingual approaches that has shown promising results in the context of spelling correction is that of Riabi *et al.* (2021), who trained `CharacterBERT` using approximately 99,000 sentences in *NArabizi* (a North African colloquial dialect written in the Latin script). Their results demonstrate that this approach yields competitive performance on the NArabizi Treebank, a testbed for noisy textual inputs, and achieves results comparable to those of models trained on much larger datasets.

Similarly, Aepli and Sennrich 2022) show that introducing character-level misspellings in high-resource source data helps improve performance on part-of-speech (POS) tagging tasks in closely related target languages: Their work focuses on language pairs from closely related branches, including Finnic, Germanic variants, and Western Romance languages.

Bernhard and Dolińska (2025) explore POS tagging robustness to misspellings in two low-resource languages, Dagur (a Mongolic language spoken by approximately 130,000 people in northern China) and Alsatian (a Germanic language from northern Europe). They fine-tune neural models on related languages and apply noise-reduction strategies, showing that the proximity between the languages has an impact on zero-shot cross-lingual performance.

6.3 Misspellings and robustness in L1 learner English

Another relevant concept in this context is *learner English*, that is, English written by speakers whose first language (L1) is not English. Although the language in question remains English, learner English embodies the linguistic influence of diverse cultural and linguistic backgrounds. Nevertheless, this perspective remains largely underrepresented in the literature on misspellings and robustness, despite its ubiquity and practical relevance in many real-world NLP applications.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Mizumoto and Nagata (2017) demonstrate that applying a spell checker (i.e., a Double-Step method, see Section 5.3) consistently improves POS tagging accuracy on texts written by native Japanese speakers. Likewise, Miaschi *et al.* (2022) investigate the impact of L1 learner errors using BERT, focusing on the CItA corpus (a collection of essays written by Italian L1 learners). Their experiments show that BERT's performance on sentence similarity tasks is variably affected depending on the error category, with misspellings having a more detrimental impact than other types of linguistic errors.

6.4 Native language identification: misspellings as an ally

A different, but closely related task is *Native Language Identification* (NLI), the task of determining a writer's first language (L1) based on their writing in a second language (typically English) Goswami *et al.* (2024).

In the context of NLI, misspellings are not merely noise but can serve as informative features reflecting transfer effects from the writer's native language. These orthographic errors often carry systematic patterns that NLP models can exploit to improve identification accuracy, making misspellings a valuable signal rather than a problem to solve in this specific task.

To the best of our knowledge, the first work to explicitly exploit the presence of misspellings in NLI was conducted by Koppel *et al.* (2005), who used a multiclass SVM model that incorporated spelling errors alongside other stylistic and structural features. Later on, Brooke and Hirst (2012) extended the approach using a larger dataset and cross-corpus evaluation, focusing on lexical features and domain adaptation techniques, though not directly modelling misspellings.

A renewed emphasis on spelling-based features arose with Chen *et al.* (2017a), who showed, using the TOEFL11 dataset, that misspellings alone could be very informative when codified as features. Markov *et al.* (2017) integrated such features in the CIC-FBK system for the NLI Shared Task, combining them with other features to further improve performance. Building on this idea, Markov *et al.* (2019) introduced orthographic features like misspelt cognates and L2-ed words-terms from the native language adapted into English orthography, while Markov *et al.* (2022) further confirmed the utility of these cues across multilingual learner corpora.

6.5 Spelling reforms

Over the years, official spelling reforms have been implemented in many languages. A spelling reform is a change in normative orthography, typically introduced by state language authorities through top-down political or institutional action. For example, the *French Conseil supérieur de la langue française* proposed a spelling reform in 1990 that affected around 2,000 words and sparked considerable public debate (Humphries 2019).[9] A similar case is the Dutch spelling reform of 2005, which also provoked widespread discussion and resistance (Nunn and Neijt 2007).

Other notable examples include the *German orthographic reform* of 1996, which introduced systematic changes (e.g., replacing "ß" with "ss" in certain cases), and various proposals for simplified spelling in English, such as those of the so-called *Simplified Spelling Board* in the early 20th century (e.g., "nite" for "night", "tho" for "though"), which, although not officially adopted, have influenced informal usage. In Spanish, the *Royal Spanish Academy* (*Real Academia Española*—RAE) has periodically introduced adjustments to spelling conventions, such as eliminating diacritical marks (e.g., in "solo") and modifying treatment of foreign words (e.g., the English term "whisky" is replaced by "güisqui").

[9] See, for example https://web.archive.org/web/20120310194816/http://www.academie-francaise.fr/langue/orthographe/plan.html (accessed 20/06/2025).

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

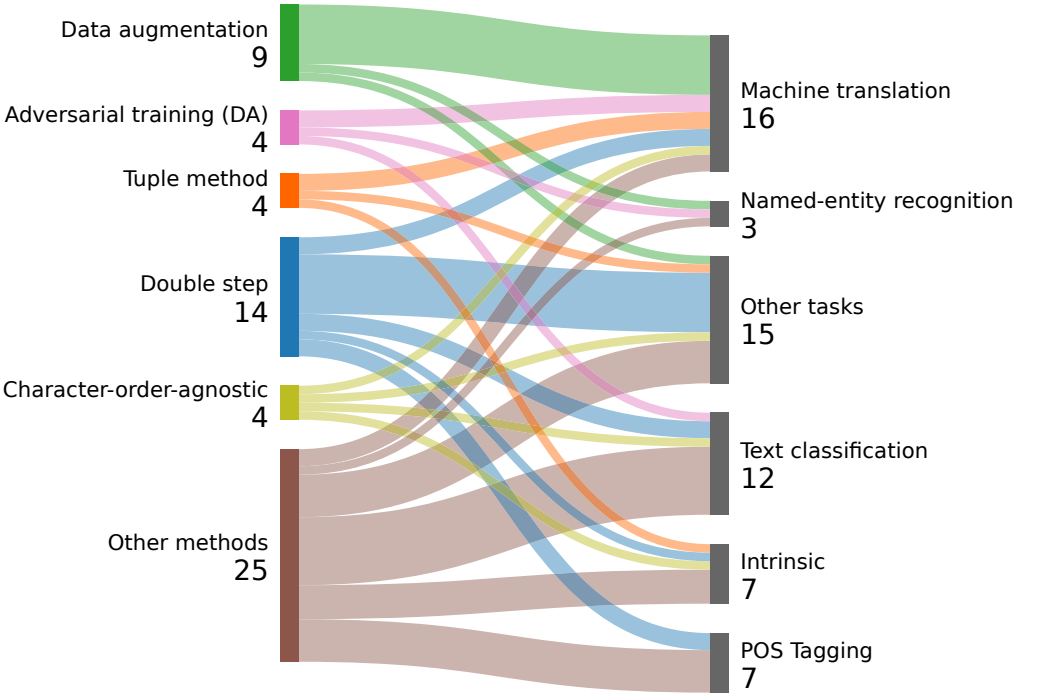

Figure 3. Distribution of methods (left) across tasks (right). Flowchart created using SankeyMATIC https://sankeymatic.com (accessed 25/08/2025).

Although not usually categorised as misspellings, these reforms can introduce variation and ambiguity in corpora, especially during transitional periods. NLP models trained on pre- or post-reform data may misclassify reformed spellings as errors or unknown words. In this context, spelling reforms represent a socio-linguistic phenomenon that can significantly affect downstream NLP tasks and serve as an additional example of the challenges systems resilient to misspellings have to cope with.

7. Tasks, evaluation metrics, and datasets

Misspellings affect written text in a broad sense, and thus, no text-related application is safe from them. However, the phenomenon has been more actively investigated in particular contexts, with machine translation and text classification being the most prolific such areas. Figure 3 gives an insight into how methods have been applied to which tasks at the time of writing this survey. In this section, we turn to describe the most important tasks (Section 7.1) in which misspellings have been investigated, by also discussing the most employed evaluation metrics (Section 7.2), dedicated events (Section 7.3), and datasets (Section 7.4).

7.1 Main tasks

The main tasks in which the phenomenon of misspellings has been more thoroughly investigated are listed below:

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

- *Machine translation* (MT) is a supervised learning task that involves producing a text in a target language that is a translation equivalent of a text written in a different source language. With most modern MT systems relying on neural networks, the field is nowadays broadly referred to as Neural Machine Translation (NMT). Undoubtedly, NMT is the field in which more methods for misspellings have been applied. One possible reason why this area has attracted a lot of attention may have to do with the appearance of two influential papers by Belinkov and Bisk (2018) and Heigold *et al.* (2018) that brought the importance of misspellings in MT to the fore. Most methods to counter misspellings in NMT rely on data augmentation (Section 5.1); see, for example, (Alam and Anastasopoulos 2020; Belinkov and Bisk 2018; Cheng *et al.* 2019; Heigold *et al.* 2018; Karpukhin *et al.* 2019; Li *et al.* 2021; Park *et al.* 2020; Passban *et al.* 2021; Salesky *et al.* 2021; Wang *et al.* 2020; Vaibhav *et al.* 2019; Zheng *et al.* 2019).
- *Text classification* (TC) is the supervised learning task of assigning class labels to unseen documents (Sebastiani 2002). While the class labels may virtually represent *anything* (e.g., from types of news to opinion stance to characteristics of an author), the most important applications of TC for the concerns of this survey include *spam filtering* and *content moderation* (Kurita *et al.* 2019). The use of misspellings in such contexts might be deliberate and malicious, targeting specific relevant words so that they become unrecognisable for an automatic detector (thus eluding any ban), but easily recognisable for the final recipient of the message. Some relevant approaches focusing in TC include (Chen *et al.* 2022; Kurita *et al.* 2019; Li *et al.* 2016; Li *et al.* 2019a; Sankar *et al.* 2021).
- *Named entity recognition* (NER) (Sang and De Meulder 2003) is the task of identifying and categorising (potentially multi-word) expressions referring to entities such as names, locations, and organisations, in text (e.g., *White House*). Namysl *et al.* (2020) observed that humans can resolve NER even in the presence of corrupted inputs and studied automatic ways for developing NER systems resilient to misspellings. Some techniques explored for NER include data augmentation (Namysl *et al.* 2020; Zhou *et al.* 2020) and contrastive learning (Chen *et al.* 2022).
- *Native language identification:* Given a speaker of two languages, the goal of Native Language Identification is to discover the L1 language of the author of a given text written in an L2 language Goswami *et al.* (2024). In this case, misspellings exhibit patterns that are often strongly correlated with the mother tongue, thus becoming allies to discover the L1 language (see also Section 6.4).
- *Other downstream tasks* tested in the field of misspellings include:
  - *Part-Of-Speech tagging* (PoST) is the task of labelling words with their correct morphosyntactic part-of-speech (e.g., verb, noun, preposition, etc.).
  - *Paraphrase detection* is the task of determining whether two given sentences have the same meaning or not.
  - *Identification of textual entailment* is the task of determining whether there is entailment, contradiction, or no relation between two given sentences.
  - *Question answering:* is the task of providing natural language answers to natural language questions.

  Influential papers that tested some of these tasks are: Doval *et al.* (2020); Jones *et al.* (2020); Malykh *et al.* (2018); Sidiropoulos and Kanoulas (2022).
- *Intrinsic tasks:* misspellings have also been tested by means of problems carefully designed to probe the capability of the system to cope with one specific language-related ability. Some of these include:

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

- *Semantic word similarity* is the task of identifying semantic relationships between words like *cat* and *dog*, or *tall* and *short* (Lenci *et al.* 2021).
- *Analogy completion*, the task of inferring which, among a closed set of options, is the more likely missing word from an incomplete analogy like *Colosseo stands to Rome as the Buckingham Palace stands to . . .*? (Lenci *et al.* 2021).
- *Outlier detection*, the task of identifying the word from a set of candidates that bears less semantic similarity to the rest of the terms in that set (Camacho-Collados and Navigli 2016).

Methods that have been assessed in intrinsic tasks include those by Chen *et al.* (2022); Doval *et al.* (2020); Edizel *et al.* (2019); Pruthi *et al.* (2019); Sperduti *et al.* (2021).

7.2 Evaluation metrics

The most straightforward way to measure the robustness of a system to the presence of misspellings (and the way most papers have indeed adhered to) comes down to simply confronting the performance a model scores with and without noisy inputs, given a standard evaluation measure for a specific task.

More formally, let $m \in \mathscr{M}$ be a generic inference model $m : \mathcal{X} \rightarrow \mathcal{Y}$ issuing predictions $\hat{y} \in \mathcal{Y}$ on textual inputs $x \in \mathcal{X}$, that has been trained to perform any given downstream task (classification, translation, etc.), and let $e : \mathscr{M} \times (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathbb{R}$ be our evaluation measure (F_1 score, BLEU score, etc.) of choice, that is any scoring function that takes as input a model and a (labelled) test set, and computes a value reflecting the empirical *goodness* of m. The degradation in performance due to the presence of misspellings can generally be estimated as the difference in performance:

$$e\left(m, \{(x_i, y_i)\}_{i=1}^n\right) - e\left(m, \{(\tilde{x}_i, y_i)\}_{i=1}^n\right)$$

where $\tilde{x}_i$ is a misspelt variant of the (clean) input x_i.

Such an evaluation strategy is generic enough to apply to virtually any supervised task, and therefore has nothing specific to do with any particular evaluation metric. Typical evaluation measures used in the tasks discussed in Section 7.1 include, among others, BLEU (Papineni *et al.* 2002) and METEOR (Banerjee and Lavie 2005) for machine translation; precision, recall, and F_β score for text classification and named entity recognition (van Rijsbergen 1979); Pearson correlation and Spearman correlation for intrinsic tasks (Doval *et al.* 2020; Lenci *et al.* 2021). An in-depth survey of these and other specific evaluation metrics is out of the scope of this article.

A few metrics have been proposed by Anastasopoulos (2019), which are particularly suitable for measuring the robustness of misspellings in the context of machine translation. These are based on the observation that any *perfectly robust-to-noise MT system would produce the exact same output for the clean and erroneous versions of the same input sentence*. The intuition is sketched as follows: let m^* be a perfect MT system; then $m^*(x)$ should produce the same (correct) prediction y^* as $m^*(\tilde{x})$, with $\tilde{x}$ a noisy variant of x. Such a perfect model is typically unavailable, but we might have a reasonably good model m instead. System robustness is therefore estimated by computing the extent to which $m(x)$ produces outputs similar to $m(\tilde{x})$, even though such predictions might not be perfect. In light of this, Anastasopoulos (2019) proposes the Robustness Percentage (RB) as:

$$\text{RB} = 100 \times \frac{|\{(x, \tilde{x}) : m(x) = m(\tilde{x})\}_{(x,\tilde{x})\in D}|}{|D|} \tag{1}$$

where D is a dataset of pairs of correct and noisy inputs.

In the same paper, Anastasopoulos (2019) propose the Target-Source Noise Ratio (NR), a more fine-grained evaluation measure that also accounts for the distance between x and $\tilde{x}$, given that

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

small differences would count just as much as large differences for RB. Instead, NR tries to factor out this distance d, which is computed using a surrogate evaluation metric like BLEU or METEOR, for example. NR is defined as:

$$\mathrm{NR}(m, x, \tilde{x}) = \frac{d(m(x), m(\tilde{x}))}{d(x, \tilde{x})} \tag{2}$$

Anastasopoulos (2019) suggest reporting the mean NR across all pairs x and $\tilde{x}$ contained in a dataset.

Michel *et al.* (2019) propose a metric for evaluating adversarial attacks that requires access to the correct translation y^*. The metric requires a similarity function s and is computed for each pair of inputs x and $\tilde{x}$ as follows:

$$A(m, x, \tilde{x}, y^*) = s(x, \tilde{x}) + \frac{s(m(x), y^*) - s(m(\tilde{x}), y^*)}{s(m(x), y^*)} \tag{3}$$

the adversarial attack is considered to be successful whenever $A(m, x, \tilde{x}, y^*) > 1$; the metric therefore computes the fraction of successful cases against all cases in a dataset.

7.3 Conferences and workshops

The most important venues, including workshops, conferences, and shared tasks, that have been devoted to discussing the problem of misspellings in NLP include:

- **Workshop on Analytics for Noisy Unstructured Text Data (AND):** This workshop had five editions run from 2007 to 2011. The main objective of AND was to gather papers discussing techniques for dealing with noisy inputs. The notion of *noisy inputs* encompasses misspellings, but also grammatical error correction, text normalisation, spelling correction, and any other form of noise affecting textual data as those generated through speech recognition systems or OCR. The first workshop was co-located in the 2007 edition of the *Joint Conference of Artificial Intelligence* (IJCAI), although no proceedings seem to be available online. The second edition was co-located at the SIGIR conference in the next year (Lopresti *et al.* 2008), followed by a third edition co-located in the *International Conference On Document Analysis and Recognition* (ICDAR) (Lopresti *et al.* 2009), and a fourth edition co-located in the *International Conference on Information and Knowledge Management* (CIKM) (Basili *et al.* 2010). The workshop then evolved as a *Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data* (Dey *et al.* 2011) and was, to the best of our knowledge, discontinued after that.
- **Robustness task at the *World Machine Translation* (WMT) Conference:** This task was first proposed in 2019 (Li *et al.* 2019b) and was later followed by a new edition in 2020 (Specia *et al.* 2020). To the best of our knowledge, this is the only shared task specifically devoted to testing the MT systems' resiliency to misspellings. In both editions, the shared tasks focused on the same language pairs: *English-French* and *English-Japanese*. In the first edition, the test set was constructed by applying the MTNT protocol (Michel and Neubig 2018) to data gathered from Reddit, while the previously existing datasets (WMT15 for the English-French pair, and KFTT (Neubig 2011), JESC (Pryzant *et al.* 2018), and TED Talks (Cettolo *et al.* 2012) for the English-Japanese pair) were employed as the training set. The systems were evaluated by professional translators as well as in terms of the BLEU score. In the second edition, the news dataset WMT20 was employed as the training set, while the test sets consisted of multiple sources, including Wikipedia and Reddit comments, among others.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

- **Workshop on Noisy and User-Generated Text (W-NUT):**[10] This is an ongoing workshop series that started in 2015, has been held every year (except 2023), with the last edition co-located at the NAACL 2025; the proceedings of all editions are published in the ACL Anthology. The workshop focuses on noise-generated content in social networks and is not exclusively devoted to misspellings. The workshop gathers papers dealing with tasks as disparate as geolocalisation prediction, global and regional trend detection and event extraction, fairness and biases in NLP models, etc.

7.4 Datasets

In this section, we turn to describe the most important types of datasets that have been used for training and evaluation of systems dealing with misspellings in literature, with particular emphasis on the techniques that have been employed for generating them. Since misspellings are relatively infrequent in real texts (with varying levels of prevalence that depend on the medium), the aim of these techniques is to guarantee a relatively high number of misspellings in the corpus, somewhat akin to oversampling strategies often used in extremely imbalanced supervised scenarios. However, as recalled from Section 3.2, the distinction between *natural* and *synthetic* misspellings is extrinsic to their surface form and lies in their mode of generation. Since the latter are created procedurally, they are easier to produce and therefore dominate the landscape of currently available datasets.

Broadly speaking, these techniques can be grouped as belonging to the following categories:

- Natural misspellings (Section 7.4.1): techniques that collect real misspellings from textual data, that is errors that occur spontaneously in user-generated (e.g., in social media, emails) or technologically-generated contexts (e.g., optical character recognition, automatic transcription).
- Artificial Misspellings (Section 7.4.2): techniques for generating synthetic misspellings out of the original (clean) words from the texts in a dataset.
- Hybrid approach (Section 7.4.3): consists of using error correction databases (i.e.,, databases in which real misspellings have been labelled with the correct word) to inject misspellings in clean texts. The approach is called hybrid since the misspellings being injected are real, but the injection itself is artificial.

7.4.1 Datasets of natural misspellings

This technique comes down to collecting real misspellings to form a dataset. Since real misspellings are relatively infrequent, datasets of natural misspellings are generated by scanning large quantities of text and retaining those entries in which some misspellings are identified.

The MTNT dataset (Michel and Neubig 2018) represents, to the best of our knowledge, the only publicly available resource devoted to collecting natural misspellings for research purposes. MTNT arises in the context of machine translation and has come to represent a reference in the field (authors such as Park *et al.* (2020); Salesky *et al.* (2021); Vaibhav *et al.* (2019); Zhou *et al.* (2019), among many others, used it as a testbed for their methods). The dataset consists of four pairs of languages (French-English, English-French, Japanese-English, and English-Japanese), and contains no less than 75,005 instances gathered from Reddit. Misspellings have been identified with the aid of text normalisation tools, word vocabularies, and scores of perplexity generated by language models as *judgments* on the feasibility of the texts given as input.

[10] https://aclanthology.org/venues/wnut/.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

7.4.2 Datasets of artificial misspellings

Since misspellings affect written natural language in general, they potentially harm *any* textual application one could think of. For this reason, when it comes to measuring the impact that misspellings cause in any downstream task, or when training models that ought to be robust to them, it is customary to simply take standard datasets routinely used for these downstream tasks and produce variants of them that contain misspelt entries; this is the approach followed by, for example Belinkov and Bisk (2018); Heigold *et al.* (2018); Passban *et al.* (2021); Sperduti *et al.* (2021). Given that the phenomenon of misspellings is orthogonal to the downstream tasks in which they are studied, we refrain from listing typical datasets customarily used across different disciplines (a glance at Table 1 reveals many of them).

The most common strategy comes down to generating synthetic misspelt variants of the original words in a text. Some techniques that have been proposed for this purpose (see, e.g., Belinkov and Bisk 2018; Kumar *et al.* 2020; Moradi and Samwald 2021) and that have been reproduced in other papers are listed below. The terminology we use for naming these types of misspellings is not standard in the literature, but is, we believe, appropriate for describing them. A list of methods with the types of misspellings they used is summarised succinctly in Table 2.

- Full Permutation: involves generating a new token by completely permuting the characters of a term, for example, *misspell* → *pseilmls*.
- Middle Permutation: generates a new token by permuting all characters of a term except the first and last, which remain in place, for example, *misspell* → *mpseisll*. This is also known as *garbling* (Sperduti *et al.* 2021) or *scrambling* (Heigold *et al.* 2018).
- Swap: consists of choosing two adjacent characters at random from a word and interchanging their positions, for example, *misspell* → *mispsell*.
- Qwerty: consists of emulating typographical errors that are likely to arise when employing a QWERTY layout, for example, *misspell* → *mi9sspell* (note the key "9" is placed nearby the key 'i' in this layout). This kind of error is a special subtype of Addition (see below).
- Addition: comes down to adding one or more characters to the target word, for example, *misspell* → *misspr*ell.
- Deletion: amounts to removing one or more characters from a word, for example, *miss*pell → *mispell*.
- Substitution: consists of choosing one character at random from a word and replacing it with another character, for example, *misspell* → *mr*sspell.

7.4.3 Datasets of hybrid misspellings

Hybrid misspellings are *natural* misspellings found somewhere else that have been *artificially* injected in a different context. These misspellings are typically taken from text normalisation and grammar correction databases, in which they are listed along with the correct surface form. Some popular examples of such databases are listed below:

- Max and Wisniewski (2010) proposed *WiCoPaCo*,[11] a French database of misspellings created with the edit correction log of Wikipedia.
- Zesch (2012) generated, also using the edit correction logs of Wikipedia, a German database of misspellings.
- Šebesta *et al.* (2017) created a database of misspellings from text generated by second-language learners in Czech.[12]

[11] https://wicopaco.limsi.fr/.
[12] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2143.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Table 2. Types of misspellings applied in each paper. We included in this table only works that used synthetic misspellings and that explicitly described the kind of misspellings used

Papers	Full Perm.	Swap	Middle Perm.	Qwerty	Addition	Deletion	Substitution
Agarwal *et al.* (2007)	×	×	×	✓	✓	✓	✓
Li *et al.* (2016)	×	×	×	×	×	×	✓
Belinkov and Bisk (2018)	✓	✓	✓	✓	×	×	×
Heigold *et al.* (2018)	×	✓	✓	×	×	×	✓
Yang and Gao (2019)	×	✓	×	×	✓	✓	×
Kurita *et al.* (2019)	✓	×	×	×	×	×	✓
Karpukhin *et al.* (2019)	×	✓	×	×	×	✓	✓
Li and Specia (2019)	×	×	×	✓	✓	✓	×
Doval *et al.* (2020)	×	×	×	×	✓	✓	✓
Kumar *et al.* (2020)	×	×	×	✓	✓	×	✓
Nguyen and Grieve (2020)	×	✓	×	✓	✓	✓	×
Jones *et al.* (2020)	×	✓	×	×	✓	✓	✓
Namysl *et al.* (2020)	✓	✓	✓	✓	×	×	×
Park *et al.* (2020)	×	×	×	×	✓	✓	✓
Moradi and Samwald (2021)	×	✓	×	×	✓	✓	✓
Sankar *et al.* (2021)	×	✓	×	×	✓	✓	×
Sperduti *et al.* (2021)	✓	×	✓	×	×	×	×
Passban *et al.* (2021)	×	×	×	×	✓	✓	✓
Wang *et al.* (2021a)	×	✓	✓	✓	✓	✓	✓
Sidiropoulos and Kanoulas (2022)	✓	✓	✓	×	✓	✓	✓
Cao *et al.* (2023)	×	×	✓	×	×	×	×
Wang *et al.* (2023)	×	✓	✓	✓	✓	✓	✓
Zhang *et al.* (2024)	×	✓	✓	✓	✓	✓	✓
Zhu *et al.* (2023)	×	✓	✓	✓	✓	✓	✓
Pan *et al.* (2024)	×	✓	×	×	✓	✓	×
Moffett and Dhingra (2025)	✓	✓	✓	✓	✓	✓	✓
Satheesh *et al.* (2025)	×	✓	✓	✓	✓	✓	✓
Pagnoni *et al.* (2025)	×	×	✓	✓	×	✓	✓
Wang *et al.* (2025)	×	×	✓	×	×	×	×

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Other spelling correction databases that could be potentially useful for creating hybrid misspellings datasets include, among many others, those by Faruqui *et al.* (2018); Hagiwara and Mita (2020); Grundkiewicz and Junczys-Dowmunt (2014); Napoles *et al.* (2017). However, to the best of our knowledge, no one before has come to use them for this purpose.

Taking a database of misspellings and a dataset specific to any downstream task as inputs, one can easily generate a variant of the dataset which contains misspellings. The process is straightforward and comes down to finding word occurrences that appear, as the correct surface form, in any entry of the database, and then replacing such a word with any of the misspelt variants recorded for it. Some popular examples of datasets created following this procedure include those by Belinkov and Bisk (2018) and Karpukhin *et al.* (2019).

7.4.4 Benchmarks

There are four main benchmarks that introduce misspellings in adversarial settings, namely AdGlue, AdGlue++, PromptRobust, and eBench, which we discuss in what follows. In each case, misspellings are generated using TextBugger (Li *et al.* 2019a).

Two of these benchmarks, AdvGlue and AdvGlue++, come from the same research team (Wang *et al.* 2021a, 2023). The idea is to use various state-of-the-art adversarial methods to create perturbed instances. These methods target models from different perspectives, including character-based, word-based, and syntax-based approaches. The adversarial methods are applied across the entire GLUE testbed. In the case of AdvGlue (Wang *et al.* 2021a), the authors focus on models like BERT, DeBERTa, and others, but exclude larger LLMs. In contrast, AdvGlue++ (Wang *et al.* 2023) tests the best-performing LLMs, including GPT-3.5, LLaMA, and GPT-4.

Another relevant benchmark that includes misspellings is PromptRobust (Zhu *et al.* 2023). The aim of this benchmark is to evaluate larger LLMs against various adversarial datasets. However, the approach does not introduce misspellings in instances or labels, but rather in the prompts given to the LLMs. Similarly, Zhang *et al.* (2024) created a benchmark, called eBench, to test the most prominent and recent LLMs with different types of adversarial attacks, including one based on misspellings. In this case, the challenging dataset used is called AlpacaEval.

Additionally, Cao *et al.* (2023) have produced a dedicated scrambled text benchmark for LLMs, called *Scrambled Bench*, which tests the resilience of LLMs to internally scrambled text in text reconstruction and question-answering tasks. In contrast to the above-discussed benchmarks, Scrambled Bench does not rely on TextBugger.

8. Large language models against misspellings

Large Language Models (LLMs) have brought about a major revolution in the field of NLP, achieving state-of-the-art performance across several tasks (Bubeck *et al.* 2023). LLMs have now become part of our everyday life with the availability of proprietary platforms like ChatGPT (OpenAI 2024), Gemini (Anil and 1376 other authors 2024), and open models like LLaMA (Touvron *et al.* 2023).

As LLMs are trained by big companies on vast amounts of data, there are no specific methods designed to *fix* or *reduce* the impact of misspellings. Instead, a number of papers focus on diagnosing and analysing how these LLMs perform in several generic tasks,[13] such as solving student tests (Puccetti *et al.* 2024), respecting morpho-syntactic constraints (Miaschi *et al.* 2024), among many others (see, e.g., Chang *et al.* 2024). Few papers, though, and only recently, have focused the evaluation study on the impact of misspellings in LLMs. Overall, these papers show that all LLMs experience a drop in performance when tested against misspellings.

[13] See also https://huggingface.co/docs/leaderboards/open_llm_leaderboard/about.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

The most important techniques to evaluate the performance of LLMs against misspellings used in the literature include *instance-based tests* and *prompt-based* tests. We discuss both types in what follows.

8.1 Instance-based tests

Instance-based tests come down to inserting misspellings into the test instances themselves. The primary approach to testing LLMs against misspellings has been through dedicated benchmarks, as those described in Section 7.4.4.

Wang *et al.* (2023, 2024) employ the AdvGLUE benchmark to evaluate the robustness of models like GPT-3.5 and GPT-4. This benchmark includes misspellings in the test instances, as outlined in Section 7.4.4.

The benchmark AdvGLUE++ (Wang *et al.* 2023) has served to show that both GPT-3.5 and GPT-4 experience a notable drop in performance when exposed to misspellings. However, Wang *et al.* (2024) found that these models are still more robust when compared to smaller models like BART-L, DeBERTa-L, and even bigger models such as `text-davinci-002`. Despite these insights, neither AdvGLUE nor AdvGLUE++ allow for a detailed ablation study, since the exact quantity and typology of misspellings in the dataset are not specified.

Pan *et al.* (2024) evaluated the robustness of large language models (LLMs) to noisy input (including misspellings) in machine translation. Specifically, the authors tested Baichuan2-7B-Chat and Baichuan2-13B-Chat for Chinese–English translation, and Qwen-7B-Chat and Qwen-14B-Chat for Indonesian–Chinese translation. The models are subjected to various types of misspellings, including both synthetic and naturally occurring misspellings. The authors found that incorporating misspellings into the prompt, as demonstrated examples, can improve model robustness. The impact of misspellings varies depending on the method used to generate them and the prompting strategy applied.

In the next section, we turn to methods that incorporate misspellings in the prompt not as demonstrations, but as genuine errors, in order to test model resiliency.

8.2 Prompt-based tests

Prompt-based tests introduce misspellings into the prompts provided to the model. Notable examples include the PromptRobust (Zhu *et al.* 2023) and eBench (Zhang *et al.* 2024) benchmarks.

In contrast to instance-based tests, these benchmarks include an ablation study, which enables further disentangling of how misspellings impact model performance. Zhu *et al.* (2023) experimented with T5-large, Vicuna, LLaMA2, UL2, ChatGPT and GPT-4 on PromptRobust, while Zhang *et al.* (2024) used LLaMA, Vicuna, GPT-3.5 and GPT-4 in eBench. Both studies concluded that LLMs experience significant performance drops when faced with misspelt prompts, though GPT-4 appears to be much more resilient than other models.

A similar conclusion was reached by Cao *et al.* (2023), who introduced ScrambledBench, a benchmark designed to test LLMs on text with internally scrambled characters—a challenge closely related to the problems addressed by the models discussed in Section 5.2. Once again, GPT-4 demonstrated notable robustness to this type of misspelling.

Building on this line of research, Wang *et al.* (2025) recently investigated the same phenomenon with the aim of identifying the factors that most influence this resilience. Specifically, they sought to disentangle the relative contributions of context and word form to LLMs' robustness against misspellings, finding that word form plays a more important role than context in reconstructing scrambled text.

Among prompt-based evaluations under misspellings, Moffett and Dhingra (2025) introduce a novel task called the *recovery task*, which assesses a model's ability to reconstruct the correct surface form given a misspelt variant of a word. To this end, they present the *Ad-Word* dataset,

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

built from the 10,000 most frequent words in the Trillion Word Corpus. Each word is perturbed using approximately nine cognitively motivated misspelling strategies (e.g., typo-based, phoneme-based, visual-based). Three experimental settings are proposed:

- *Prediction without context*: Several language models, including open-source and commercial versions, were evaluated on isolated word recovery. Surprisingly enough, GPT-4 surpassed the accuracy of the human expert baseline, consisting of five annotators.
- *Prediction with context*: LLaMA2 and Mistral were tested on the same task with the aid of sentence-level context. Contextual information improved recovery in some cases but degrades performance in others, depending on the nature of the misspelling (e.g., in LLaMA2-7B, several additional visual and typographic misspellings were recovered with the aid of context, but accuracy also dropped by about 15 per cent in cases where the model had performed well without it).
- *Hate speech detection*: Words in the HOT (*Hate, Offensive, Toxic*) dataset were misspelt at varying levels. Both LLaMA2 and Mistral exhibit degraded classification performance, with the latter being more adversely affected.

The results suggest that open-source models are generally more vulnerable to misspellings than commercial counterparts.

9. Applications

Applications of systems robust to misspellings span the entire spectrum of text-based applications, with no exception. It is not our intention to list any possible such application here, but instead, highlight those in which the presence of misspellings might be of particular relevance. This is not to say that research in other applicative areas can simply disregard the problem; certainly, the presence of misspellings harms performance no matter the task, and it is worth investing efforts in trying to devise (maybe application-dependent) ways for countering them. However, in some applications, the presence of misspellings may carry over stronger implications. In particular, when the misspellings are *intentional*, that is, not due to an unadvertised typographical error. Examples of this include:

- Content moderation: language used in social networks is often informal and rich in misspellings and ungrammatical sentences; the phenomenon is well covered by Baldwin *et al.* (2013). This poses obvious difficulties for any automated analysis tool, and this is of particular concern when the misspellings are intentionally placed to escape the control of a content moderation tool. Malicious users can cover up offensive comments by means of misspellings of graphical type (e.g., those based on replacing some characters with others that are graphically similar, like replacing an "i" with "<") in order to sneak in toxic comments (e.g., "n<gger", "<d<ot") into a debate; see, for example the work by Hosseini *et al.* (2017); Kurita *et al.* (2019). As a matter of fact, in recent years, online communities have started to develop some *alternative slang* to avoid censorship that has later come to be known as *algospeak*.
  The phenomenon has had a big impact on media, to the point that it has been echoed by renowned newspapers such as the Washington Post.[14] The phenomenon is far from new, however, and we might trace its influence back to the usage of the so-called *aesopian* languages (encrypted forms of language that became popular in totalitarian regimes, see also Loseff 1984). Yet another related area in which (intentional) misspellings play a special role is that of pro-eating-disorders (pro-ED) communities, in which some users might

[14] https://www.washingtonpost.com/technology/2022/04/08/algospeak-tiktok-le-dollar-bean/.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

resort to complex lexical variants to promote disordered eating habits that may eventually lead to anorexia or obesity (Chancellor *et al.* 2016).

- Spam filtering: Span filters are text classification tools aimed at preventing the delivery of unsolicited and even potentially virus-infected emails. The use of misspellings, among many other malicious practices (Fumera *et al.* 2006), is one way for eluding the filter to reach—much to her regret—the final user (Aldwairi and Flaifel 2012; Ahmed and Mithun 2004; Lee and Ng 2005; Renuka and Hamsapriya 2010).
- Authorship analysis: Systems resilient to misspellings by design are relevant in authorship analysis. Specifically, some misspellings reveal the nativity of their author. For example, Berti *et al.* (2023) note that *de*, which is the misspelt form of *the*, is a typical phonological typo of Spanish native speakers. In a similar vein, the misspelling *to allñow* (a QWERTY error of *to allow*) would carry on potential clues about the nationality of the author, since the Spanish layout he/she is probably using places character *ñ* just to the right of character *l*. Double-step methods that clean the text as a pre-processing step are thus potentially harmful for authorship analysis endeavours. Indeed, Stamatatos (2009) cite misspellings as relevant features for authorship analysis.
- The applicative area that has, by far, received more attention with respect to the phenomenon of misspellings is machine translation (see Section 7.1). Belinkov and Bisk (2018) and Heigold *et al.* (2018) were the first to argue that neural machine translation models are heavily affected by the presence of misspellings, in contrast to human translators who have the cognitive ability to bypass misspelt entries without effectively penalising comprehension (Rayner *et al.* 2006).

10. Frontiers and the road ahead

While humans can easily read and understand misspelt text, computers still cannot, although significant developments should be noted in the world of LLMs (Section 8). No textual application is out of reach for the potential harm of misspellings (Sections 4 and 9). Despite this, the field of misspelling resiliency has attracted uneven attention, with machine translation being the only field in which the phenomenon has been more thoroughly studied, followed by text classification. This may be fostered by the lack of common definitions and frameworks concerning the concept of misspellings. Furthermore, the absence of a standardised benchmark for evaluating robustness to misspellings currently limits cross-study comparability; while several benchmarks have been proposed in the literature (see Section 7.4.4), these are typically tailored to individual studies and do not provide a unified evaluation arena. We therefore believe that addressing this gap would represent an important avenue for future research.

In any case, and even though a fine-grained quantitative comparison across model families remains technically challenging, a clear trend emerging from our review is that commercial LLMs tend to exhibit higher resilience compared to both traditional NLP models and open-source LLMs. However, while the exact mechanisms underlying this robustness are not always transparent, we hypothesise that it may result from large-scale exposure to noisy web data and perhaps sophisticated (though undisclosed) data augmentation or instruction-tuning pipelines. In this context, evaluation studies assessing robustness to misspellings across different systems and use cases may become increasingly valuable, as they provide the scientific community with insight into model behaviour that would otherwise remain inaccessible.

Another important line of research concerns the treatment of intentional, human-generated misspellings. As hinted throughout this survey, intentional misspellings pose a greater threat than unintentional misspellings (which are stochastically distributed and often affect semantically negligible tokens) since intentional obfuscations specifically target high-value keywords in order to evade moderation or content filters. As a result, these adversarial perturbations are particularly

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

effective at bypassing hate speech detection (Röttger *et al.* 2021) or LLM-generated text detection (Creo and Pudasaini 2025). This directly connects with the emerging, yet under-explored, domain of NLP security, where adversarial training techniques are increasingly studied. We therefore anticipate that focused research on misspelling-based adversarial attacks will play a key role in developing robust and secure NLP systems.

Finally, yet another important gap concerns the scarcity of non-English languages, and especially of languages other than Western ones. In the literature, there have been few studies analysing the impact of misspellings in multilingual contexts. Filling these gaps would probably help boost research in the field.

It is our impression that resiliency to misspellings should become a *native* feature of modern NLP systems, which will contribute to paving the way toward achieving significant goals:

- Models dealing with misspellings represent a cornerstone not only for handling textual errors (OCR noise, social-media typo-ridden content, etc.) but also for handling the untamed evolution of natural language, which is closely tied to the evolution of human culture. Models that resolve misspellings by analysing the context in which these appear might prove resistant to changes in morphology over time (diachronically), across different locations (diatopically), or in specific social contexts (diastratically).
- Models handling misspellings can inspire ways for attaining more efficient representations. The fact that most misspellings go unnoticed by human readers seems to suggest that the way we process them makes few distinctions between the misspelling and the clean word. From an information-theoretic point of view, this is equivalent to avoiding explicitly codifying information that carries over no really useful information, like the internal order of characters in certain words (Section 5.2) or the graphical differences between certain characters (Section 9). Put otherwise, systems resilient to misspellings should, in principle, be able to compress any spurious information.

The presence of misspellings is pervasive and affects nearly all applications of NLP. The problem spans multiple layers of complexity from different viewpoints, including linguistic, sociolinguistic, cognitive, and computational perspectives, and is far from being solved. Future directions may ideally encompass the broader dimension of language variation (e.g., genre, register, dialect) and be tailored to specific computational tasks, rather than attempting a "one-size-fits-all" solution. We hope this survey has drawn attention to the challenge of misspelling resilience and will serve as a valuable resource for researchers interested in this area.

Glossary

Adversarial attack: An adversarial attack on an artificial intelligence model aimed at undermining its capabilities in order to compromise its stability and security.

Algospeak: A deliberately coded language used by users to bypass social media censorship.

Artificial Misspelling: Artificial misspellings, also known as "synthetic noise," are misspellings that are generated by an algorithm to imitate natural misspellings. They are designed to simulate the types of errors commonly found in real-world text data. Artificial misspellings are widely used in the field of NLP, as discussed in Section 4.1, and are considered one of the most prevalent forms of noise (Belinkov and Bisk, 2018).

BPE: Byte-Pair Encoding is a data compression technique that replaces recurring sequences of characters with new tokens. In NLP, BPE is used to encode words as sequences of subword units, allowing for flexible representation of rare and unseen words. BPE is commonly applied in tasks like machine translation and text generation to improve model efficiency and vocabulary handling.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Character-based: An encoding technique used to represent words based on their sequential character composition. Character-based models use characters as the building blocks of the representation, instead of entire words. Useful for handling morphologically rich languages and capturing fine-grained information at the character level.

Error: A generic linguistic term used to describe an unsuccessful piece of language, such as a misspelling or a grammatical mistake. In the context of NLP, errors refer to deviations from the intended or correct form of text. Errors can occur due to various factors, including typos, transcription noise, or other forms of linguistic variability.

Fine-Tuning: The process of performing additional training epochs on a pre-trained (language) model, such as BERT or RoBERTa, on domain-specific data. Fine-tuning allows the model to learn task-specific patterns and improve its performance on the target domain.

Garbling: Refers to the reproduction of a text in a confused or distorted manner, as in *wrod ebmneddigs ecndoe semnacits*. In the context of NLP, garbling can occur due to various factors, including misspellings, typographical errors, or text corruption during transmission or processing. It can affect the readability and interpretation of the text, making it challenging for NLP systems to handle. Somehow, surprisingly, humans are less affected by this type of error if the first and last characters stay in place.

General linguistics: General linguistics is the discipline that studies human language in itself.

Intentional misspelling: A non-standard spelling of a word deliberately chosen by the author in order to achieve communicative advantages, such as avoiding online censorship or being part of a linguistic koiné.

LSH: Local Sensitive Hashing is an encoding technique used to reduce the dimensionality of sparse vectors. LSH groups similar items into the same *bucket* or index, allowing for efficient nearest neighbour search and similarity-based retrieval. LSH is commonly used in tasks like approximate nearest neighbour search and data deduplication.

Morphology: The study and description of the internal structure and forms of words in a language. Morphology is concerned with analysing how words are formed from smaller meaningful units called *morphemes* and how they inflect and change to convey grammatical information. Understanding morphology is important in NLP for tasks like word segmentation, lemmatisation, and morphological analysis, among others.

Natural misspelling: In the context of systems robust to misspellings, a natural misspelling is a misspelling that occurs in real-world data sources. Sources prone to this type of misspelling include social networks, OCR (Optical Character Recognition) data, or other forms of user-generated content.

Nnoise: Any unwanted alteration of the original textual source. In NLP, noise often encompasses various forms of errors or inconsistencies in text, such as typographical errors, grammatical mistakes, or other unintended linguistic variations. These alterations can arise from issues in transcription, data transmission, or human error, ultimately affecting the accuracy of language processing tasks.

OOV: Out-Of-Vocabulary. In NLP, OOV words refer to terms that are not included in a model's training dataset or vocabulary. When processing text, vocabulary-based NLP systems may encounter OOV words and struggle to generate accurate representations or predictions for them

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

due to a lack of prior information. Effectively managing OOV words is crucial to enhance the coverage and overall performance of NLP models.

Perturbation: The process of deliberately introducing modifications or disturbances to an instance of data. In NLP, perturbation involves altering a text sample by adding noise, introducing misspellings, or making other modifications. Perturbed instances are commonly used to create adversarial examples, which help evaluate the robustness of NLP models against different types of input manipulation and unexpected variations.

Psycholinguistics: Psycholinguistics is the discipline that studies the relationship between language and the mind.

Qwerty: Refers to the standard keyboard layout for the Latin alphabet, named after the arrangement of its first six letters. The QWERTY layout is widely used in English-speaking countries and serves as the default on many devices. In NLP, the term is sometimes associated with misspellings or linguistic variations that arise from typing errors commonly made on this layout.

Robustness: In NLP, robustness refers to a system's ability to effectively process text containing misspellings, noise, or other linguistic variations while maintaining reliable performance and accuracy. A robust NLP model can withstand challenging or imperfect inputs, making it essential for handling real-world text data, which frequently includes misspellings and other forms of noise.

Sociolinguistics: Sociolinguistics is the discipline that studies the relationship between language and society.

Source sentence: In machine translation, a source sentence is a sentence that is written in the source language and serves as the input for translation into a target language.

Spelling variation: A variation in spelling from the normative grammar. In our survey, it always falls under the umbrella term *misspelling*; however, it is a complex term that could also refer to entire speaker koinés or even to dialectal forms.

Target sentence: In machine translation, a target sentence is a sentence in the dataset that represents the intended translation of a corresponding source sentence, written in the target language.

Synthetic misspelling: A misspelling that is artificially generated by an algorithm and inserted in a text. Synthetic misspellings are commonly used in the context of systems robust to misspellings to simulate different types and levels of misspelt text. By introducing controlled misspellings, NLP models can be trained and evaluated to improve their robustness and generalisation to handle various forms of misspelt input.

Acknowledgements. We are grateful to the four anonymous reviewers, whose insightful comments and suggestions have greatly helped to enhance the quality and scope of this survey. This work has been supported by the project "Word Embeddings: From Cognitive Linguistics to Language Engineering, and Back" (WEMB), funded by the Italian Ministry of University and Research (MUR) under the PRIN 2022 funding scheme (CUP B53D23013050006).

Declaration of generative AI and AI-assisted technologies in the writing process. During the preparation of this work, the authors used ChatGPT in order to proofread the paper. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

Aepli N. and **Sennrich R.** (2022). Improving zero-shot cross-lingual transfer between closely related languages by injecting character-level noise. In Muresan, S., Nakov P. and Villavicencio A. (eds), *Findings of the Association for Computational Linguistics: ACL*. Dublin, Ireland: Association for Computational Linguistics, pp. 4074–4083.

Agarwal S., **Godbole S.**, **Punjani D.** and **Roy S.** (2007). How much noise is too much: A study in automatic text classification. In *Proceedings of the 7th International Conference on Data Mining (ICDM 2007)*, Omaha, US, pp. 3–12.

Ahmed S. and **Mithun F.** (2004). Word stemming to enhance spam filtering. In *Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS 2004)*, Mountain View, US.

Alam M. M. and **Anastasopoulos A.** (2020). Fine-tuning MT systems for robustness to second-language speaker variations. In *Proceedings of the 6th Workshop on Noisy User-Generated Text (NUT 2020)*, Online Event, pp. 149–158.

Aldwairi M. and **Flaifel Y.** (2012). Baeza-Yates and Navarro approximate string matching for spam filtering. In *Proceedings of the 2nd International Conference on Innovative Computing Technology (INTECH 2012)*, Stanford, US, pp. 16–20.

Anastasopoulos A. (2019). An analysis of source-side grammatical errors in NMT. In *Proceedings of the 2019 Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP 2019)*, Firenze, IT, pp. 213–223.

Andrews S. (1996). Lexical retrieval and selection processes: Effects of transposed-letter confusability. *Journal of Memory and Language* **35**(6), 775–800.

Anil R. (2024). And 1376 other authors, Gemini: A family of highly capable multimodal models. arXiv: 2312.11805 [cs.CL].

Baldwin T., **Cook P.**, **Lui M.**, **MacKinlay A.** and **Wang L.** (2013). How noisy social media text, how diffrnt social media sources? In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013)*, Nagoya, JP, pp. 356–364.

Banerjee S. and **Lavie A.** (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 (MTSE 2005)*, Prague, CZ, pp. 65–72.

Basili R., **Lopresti D. P.**, **Ringlstetter C.**, **Roy S.**, **Schulz K. U.** and **Subramaniam L.** (2010). Proceedings of the 4th Workshop on Analytics for Noisy Unstructured Text Data (AND 2010). Toronto, CA.

Belinkov Y. and **Bisk Y.** (2018). Synthetic and natural noise both break neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*, Vancouver, CA, pp. 1–13.

Benamar A., **Grouin C.**, **Bothua M.** and **Vilnat A.** (2022). Evaluating tokenizers impact on OOVs representation with transformers models. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022)*, Marseille, FR, pp. 4193–4204.

Bernhard D. and **Dolińska J.** (2025). Managing noise in part-of-speech tagging for extremely low-resource languages: Comparing strategies for corpus collection and annotation in Dagur and alsatian. *Corpus* **26**, 1–16.

Berti B., **Esuli A.** and **Sebastiani F.** (2023). Unravelling interlanguage facts via explainable machine learning. *Digital Scholarship in the Humanities* **38**(3), 953–977.

Bhargava P., **Spasojevic N.** and **Hu G.** (2017). Lithium NLP: A system for rich information extraction from noisy user-generated text on social media. In *Proceedings of the 3rd Workshop on Noisy User-Generated Text (NUT 2017)*, Copenhagen, DK, pp. 131–139.

Blair C. R. (1960). A program for correcting spelling errors. *Information and Control* **3**(1), 60–67.

Bojanowski P., **Grave E.**, **Joulin A.** and **Mikolov T.** (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5**, 135–146.

Brooke J. and **Hirst G.** (2012). Robust, lexicalized native language identification, Organizing Committee. In Kay M. and Boitet C. (eds), *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, pp. 391–408.

Bryant C., **Yuan Z.**, **Qorib M.R.**, **Cao H.**, **Ng H.T.** and **Briscoe T.** (2023). Grammatical error correction: A survey of the state of the art. *Computational Linguistics* **49**(3), 643–701.

Bubeck S., **Chandrasekaran V.**, **Eldan R.**, **Gehrke J.**, **Horvitz E.**, **Kamar E.**, **Lee P.**, **Lee Y. T.**, **Li Y.**, **Lundberg S. M.**, **Nori H.**, **Palangi H.**, **Ribeiro M. T.** and **Zhang Y.** (2023). Sparks of artificial general intelligence: Early experiments with GPT-4.

Bulté B. and **Tezcan A.** (2019). Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019)*, Firenze, IT, pp. 1800–1809.

Camacho-Collados J. and **Navigli R.** (2016). Find the word that does not belong: A framework for an intrinsic evaluation of word vector representations. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP (RepEval 2016)*, Berlin, DE, pp. 43–50.

Cao Q., **Kojima T.**, **Matsuo Y.** and **Iwasawa Y.** (2023). Unnatural error correction: GPT-4 can almost perfectly handle unnatural scrambled text. In Bouamor H., Pino J. and Bali K. (eds), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*. Association for Computational Linguistics, pp. 8898–8913.

Cettolo M., **Federico M.**, **Specia L.** and **Way A.** (2012). WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT 2012)*, Trento, IT, pp. 261–268.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Chancellor S., **Pater J. A.**, **Clear T.**, **Gilbert E.** and **De Choudhury M.** (2016). # thyghgapp: Instagram content moderation and lexical variation in pro-eating disorder communities. In *Proceedings of the 19th Conference on Computer-Supported Cooperative Work and Social Computing (CSCW 2016)*, San Francisco, US, pp. 1201–1213.

Chang Y., **Wang X.**, **Wang J.**, **Wu Y.**, **Yang L.**, **Zhu K.**, **Chen H.**, **Yi X.**, **Wang C.**, **Wang Y.**, **Ye W.**, **Zhang Y.**, **Chang Y.**, **Yu P. S.**, **Yang Q.** and **Xie X.** (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* **15**(3), 45.

Chen L., **Strapparava C.** and **Nastase V.** (2017a). Improving native language identification by using spelling errors. In Barzilay R. and Kan M. Y. (eds), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 542–546.

Chen L., **Varoquaux G.** and **Suchanek F.** (2022). Imputing out-of-vocabulary embeddings with LOVE makes language models robust with little cost. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)* **1**, 3488–3504.

Chen Q., **Zhu X.**, **Ling Z.**, **Wei S.**, **Jiang H.** and **Inkpen D.** (2017b). Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*. Vancouver, CA: Association for Computational Linguistics, pp. 1657–1668.

Cheng Y., **Jiang L.** and **Macherey W.** (2019). Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019)*, Firenze, IT, pp. 4324–4333.

Collobert R., **Weston J.**, **Bottou L.**, **Karlen M.**, **Kavukcuoglu K.** and **Kuksa P.** (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research* **12**, 2493–2537.

Crawford M., **Khoshgoftaar T. M.**, **Prusa J. D.**, **Richter A. N.** and **Al Najada H.** (2015). Survey of review spam detection using machine learning techniques. *Journal of Big Data* **2**, 1–24.

Creo A. and **Pudasaini S.** (2025). SilverSpeak: Evading AI-generated text detectors using homoglyphs. In Alam F., Nakov P., Habash N., Gurevych I., Chowdhury S., Shelmanov A., Wang Y., Artemova E., Kutlu M. and Mikros G. (eds), *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*. International Conference on Computational Linguistics. pp. 1–46.

Damerau F. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM* **7**(3), 171–176.

Devlin J., **Chang M.**, **Lee K.** and **Toutanova K.** (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 (NAACL-HLT 2019)*, Minneapolis, US, pp. 4171–4186.

Dey L., **Govindaraju V.**, **Lopresti D. P.**, **Natarajan P.**, **Ringlstetter C.** and **Roy S.** (eds). (2011). *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data (MOCR/AND 2011)*, Bejing, CN.

Doval Y., **Vilares J.** and **Gómez-Rodríguez C.** (2020). Towards robust word embeddings for noisy texts. *Applied Sciences* **10**(19), 6893.

Edizel B., **Piktus A.**, **Bojanowski P.**, **Ferreira R.**, **Grave E.** and **Silvestri F.** (2019). Misspelling-oblivious word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, Minneapolis, US, pp. 3226–3234.

Faruqui M., **Pavlick E.**, **Tenney I.** and **Das D.** (2018). WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, Brussels, BE, pp. 305–315.

Fernández E. M. and **Cairns H. S.** (2010). *Fundamentals of Psycholinguistics*. Chichester, UK: John Wiley & Sons.

Fumera G., **Pillai I.** and **Roli F.** (2006). Spam filtering based on the analysis of text information embedded into images. *Journal of Machine Learning Research* **7**(12), 2699–2720.

Goodfellow I. J., **Shlens J.** and **Szegedy C.** (2015). Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, US, pp. 1–11.

Goswami D., **Thilagan S.**, **North K.**, **Malmasi S.** and **Zampieri M.** (2024). Native language identification in texts: A survey. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3149–3160.

Grundkiewicz R. and **Junczys-Dowmunt M.** (2014). The WikEd error corpus: A corpus of corrective Wikipedia edits and its application to grammatical error correction. In *Proceedings of the 9th International Conference on NLP (PolTAL 2014)*, Warsaw, PL, pp. 478–490.

Hagiwara M. and **Mita M.** (2020). Github typo corpus: A large-scale multilingual dataset of misspellings and grammatical errors. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, Marseille, FR, pp. 6761–6768.

He K., **Chen X.**, **Xie S.**, **Li Y.**, **Dollár P.** and **Girshick R. B.** (2022). Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, New Orleans, US, pp. 15979–15988.

Healy A. F. (1976). Detection errors on the word the: Evidence for reading units larger than letters. *Journal of Experimental Psychology: Human Perception & Performance* **2**(2), 235–242.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Heigold G., **Varanasi S.**, **Neumann G.** and **van Genabith J.** (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 (AMTA 2018)*, Boston, US, pp. 68–80.

Hládek D., **Staš J.** and **Pleva M.** (2020). Survey of automatic spelling correction. *Electronics* **9**(10), 1670.

Hosseini H., **Kannan S.**, **Zhang B.** and **Poovendran R.** (2017). Deceiving Google's perspective API built for detecting toxic comments. arXiv preprint arXiv: 1702.08138.

Humphries E. (2019). # Jesuiscirconflexe: The French spelling reform of 1990 and 2016 reactions. *Journal of French Language Studies* **29**(3), 305–321.

James C. (2013). *Errors in Language Learning and use: Exploring Error Analysis*. New York, NY: Routledge.

Jones E., **Jia R.**, **Raghunathan A.** and **Liang P.** (2020). Robust encodings: A framework for combating adversarial typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, Online Event, pp. 2752–2765.

Joulin A., **Grave E.**, **Bojanowski P.** and **Mikolov T.** (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, Valencia, ES, pp. 427–431.

Karpukhin V., **Levy O.**, **Eisenstein J.** and **Ghazvininejad M.** (2019). Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-Generated Text (NUT 2019)*, Hong Kong, CN, pp. 42–47.

Kim Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, (EMNLP 2014)*, Doha, QA, pp. 1746–1751.

Koppel M., **Schler J.** and **Zigdon K.** (2005). Determining an author's native language by mining a text for errors. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD '05*, New York, NY, USA: Association for Computing Machinery, pp. 624–628.

Krizhevsky A., **Sutskever I.** and **Hinton G. E.** (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS 2012)*, Lake Tahoe, US, pp. 1106–1114.

Kudo T. and **Richardson J.** (2018). SentencePiece: A simple and language-independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, Brussels, BE, pp. 66–71.

Kumar A., **Makhija P.** and **Gupta A.** (2020). Noisy text data: Achilles' heel of BERT. In *Proceedings of the 6th Workshop on Noisy User-Generated Text (NUT 2020)*, Online Event, pp. 16–21.

Kurita K., **Belova A.** and **Anastasopoulos A.** (2019). Towards robust toxic content classification, arXiv: 1912.06872 [cs.CL].

Lee H. and **Ng A. Y.** (2005). Spam deobfuscation using a hidden Markov model. In *Proceeding of the 2nd Conference on Email and Anti-Spam (CEAS 2005)*, Stanford, US, pp. 1–8.

Lenci A., **Sahlgren M.**, **Jeuniaux P.**, **Gyllensten A.C.** and **Miliani M.** (2021). A comprehensive comparative evaluation and analysis of distributional semantic models, arXiv: 2105.09825 [cs.CL].

Li J., **Ji S.**, **Du T.**, **Li B.** and **Wang T.** (2019a). Textbugger: Generating adversarial text against real-world applications. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS 2019)*, San Diego, US, pp. 1–15.

Li X., **Michel P.**, **Anastasopoulos A.**, **Belinkov Y.**, **Durrani N.**, **Firat O.**, **Koehn P.**, **Neubig G.**, **Pino J. M.** and **Sajjad H.** (2019b). Findings of the first shared task on machine translation robustness. In *Proceedings of the 4th Conference on Machine Translation (WMT 2019)*, Firenze, IT, pp. 91–102.

Li Y., **Cohn T.** and **Baldwin T.** (2016). Learning robust representations of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, Austin, US, pp. 1979–1985.

Li Z., **Rei M.** and **Specia L.** (2021). Visual cues and error correction for translation robustness. In *Findings of the Association for Computational Linguistics @ EMNLP 2021*. Punta Cana, DO, pp. 3153–3168.

Li Z. and **Specia L.** (2019). Improving neural machine translation robustness via data augmentation: Beyond back-translation. In *Proceedings of the 5th Workshop on Noisy User-Generated Text (NUT 2019)*, Hong Kong, CN, pp. 328–336.

Liu N. F., **Schwartz R.** and **Smith N. A.** (2019). Inoculation by fine-tuning: A method for analyzing challenge datasets, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, Minneapolis, MN, US, pp. 2171–2179.

Ljubesic N., **Erjavec T.** and **Fiser D.** (2017). Adapting a state-of-the-art tagger for South Slavic languages to non-standard text. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing, BSNLP@EACL 2017*, Valencia, Spain: Association for Computational Linguistics, pp. 60–68,

Lopresti D. P., **Roy S.**, **Schulz K. U.** and **Subramaniam L.** (eds) (2008). *Proceedings of the 2nd Workshop on Analytics for Noisy Unstructured Text Data (AND 2008)*, Singapore, SG, pp. 1–118.

Lopresti D. P., **Roy S.**, **Schulz K. U.** and **Subramaniam L.** (eds). (2009). *Proceedings of the 3rd Workshop on Analytics for Noisy Unstructured Text Data (AND 2009)*, Barcelona, ES, pp. 1–124.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Loseff L. (1984). *On the Beneficence of Censorship: Aesopian Language in Modern Russian Literature*. Bristol, UK: Peter Lang International Academic Publishers.

Malykh V., **Logacheva V.** and **Khakhulin T.** (2018). Robust word vectors: Context-informed embeddings for noisy texts. In *Proceedings of the 4th Workshop on Noisy User-Generated Text (NUT 2018)*, Brussels, BE, pp. 54–63.

Mamta Ahmad Z. and **Ekbal A.** (2023). Elevating code-mixed text handling through auditory information of words. In *Proceedings of the 2023 Conference On Empirical Methods in Natural Language Processing, EMNLP 2023*, Association for Computational Linguistics, pp. 15918–15932, Singapore.

Markov I., **Chen L.**, **Strapparava C.** and **Sidorov G.** (2017). CIC-FBK approach to native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 374–381.

Markov I., **Nastase V.** and **Strapparava C.** (2019). Anglicized words and misspelled cognates in native language identification. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 275–284.

Markov I., **Nastase V.** and **Strapparava C.** (2022). Exploiting native language interference for native language identification. *Natural Language Engineering* **28**(2), 167–197.

Max A. and **Wisniewski G.** (2010). Mining naturally-occurring corrections and paraphrases from Wikipedia's revision history. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Valletta, MT, pp. 3143–3148.

Mays E., **Damerau F. J.** and **Mercer R. L.** (1991). Context based spelling correction. *Information Processing & Management* **27**(5), 517–522.

McCusker L. X., **Gough P. B.** and **Bias R. G.** (1981). Word recognition inside out and outside in. *Journal of Experimental Psychology: Human Perception & Performance* **7**(3), 538–551.

Miaschi A., **Brunato D.**, **Dell'Orletta F.** and **Venturi G.** (2022). On robustness and sensitivity of a neural language model: A case study on italian l1 learner errors. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **31**, 426–438.

Miaschi A., **Dell'Orletta F.** and **Venturi G.** (2024). Evaluating large language models via linguistic profiling. In Al-Onaizan Y., Bansal M. and Chen Y. (eds), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP, address = Miami, US*. Association for Computational Linguistics, pp. 2835–2848.

Michel P., **Li X.**, **Neubig G.** and **Pino J. M.** (2019). On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, Minneapolis, US, pp. 3103–3114.

Michel P. and **Neubig G.** (2018). MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, Brussels, BE, pp. 543–553.

Mikolov T., **Chen K.**, **Corrado G.** and **Dean J.** (2013). Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR 2013), Workshop Track Proceedings*, Scottsdale, US, pp. 1–12.

Mizumoto T. and **Nagata R.** (2017). Analyzing the impact of spelling errors on pos-tagging and chunking in learner english. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (nlptea 2017)*, pp. 54–58.

Moffett L. and **Dhingra B.** (2025). Close or cloze? assessing the robustness of large language models to adversarial perturbations via word recovery. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025*, Abu Dhabi, UAE: Association for Computational Linguistics, pp. 6999–7019.

Moradi M. and **Samwald M.** (2021). Evaluating the robustness of neural language models to input perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, Punta Cana, DO, pp. 1558–1570.

Muñoz-Ortiz A., **Blaschke V.** and **Plank B.** (2025). Evaluating pixel language models on non-standardized languages. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025*, Abu Dhabi, UAE: Association for Computational Linguistics, pp. 6412–6419.

Naik A., **Ravichander A.**, **Sadeh N. M.**, **Rosé C. P.** and **Neubig G.** (2018). Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, Santa Fe, New Mexico, USA, pp. 2340–2353.

Namysl M., **Behnke S.** and **Köhler J.** (2020). NAT: Noise-aware training for robust neural sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, Online Event, pp. 1501–1517.

Náplava J., **Popel M.**, **Straka M.** and **Straková J.** (2021). Understanding model robustness to user-generated noisy texts. In *Proceedings of the 7th Workshop on Noisy User-Generated Text (NUT 2021)*, Online Event, pp. 340–350.

Napoles C., **Sakaguchi K.** and **Tetreault J. R.** (2017). JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, Valencia, ES, pp. 229–234.

Neubig G. (2011). The Kyoto free translation task. *Available at:* http://www.phontron.com/kftt.

Nguyen D. and **Grieve J.** (2020). Do word embeddings capture spelling variation? In *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*, Barcelona, ES, pp. 870–881.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Nunn A. and **Neijt A.** (2007). The recent history of dutch orthography (ii). problems solved and created by the 2005 reform.
OpenAI. (2024). GPT-4 technical report. arXiv: 2303.08774 [cs.CL].
Pagnoni A., **Pasunuru R.**, **Rodriguez P.**, **Nguyen J.**, **Muller B.**, **Li M.**, **Zhou C.**, **Yu L.**, **Weston J. E.**, **Zettlemoyer L.**, **Ghosh G.**, **Lewis M.**, **Holtzman A.** and **Iyer S.** (2025). Byte latent transformer: Patches scale better than tokens. In Che W., Nabende J., Shutova E. and Pilehvar M. T. (eds), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vienna, Austria: Association for Computational Linguistics, pp. 9238–9258.
Pan L., **Leng Y.** and **Xiong D.** (2024). Can large language models learn translation robustness from noisy-source in-context demonstrations? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024*, Torino, Italy, pp. 2798–2808.
Papineni K., **Roukos S.**, **Ward T.** and **Zhu W.** (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadelphia, US, pp. 311–318.
Parikh A. P., **Täckström O.**, **Das D.** and **Uszkoreit J.** (2016). A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, Austin: US. The Association for Computational Linguistics, pp. 2249–2255.
Park J., **Sung M.**, **Lee J.** and **Kang J.** (2020). Adversarial subword regularization for robust neural machine translation. In *Findings of the Association for Computational Linguistics @ EMNLP 2020*, Online Event, pp. 1945–1953.
Passban P., **Saladi P. S. M.** and **Liu Q.** (2021). Revisiting robust neural machine translation: A transformer case study. In *Findings of the Association for Computational Linguistics @ EMNLP 2021*. Punta Cana, DO, pp. 3831–3840.
Plank B. (2016). What to do about non-standard (or non-canonical) language in NLP. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, Bochum, DE, pp. 13–20.
Pruthi D., **Dhingra B.** and **Lipton Z. C.** (2019). Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019)*, Firenze, IT, pp. 54–63.
Pryzant R., **Chung Y.**, **Jurafsky D.** and **Britz D.** (2018). JESC: Japanese-English subtitle corpus. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, JP, pp. 1113–1117.
Puccetti G., **Cassese M.** and **Esuli A.** (2024). The invalsi benchmark: Measuring language models mathematical and language understanding in Italian.
Ravichander A., **Dalmia S.**, **Ryskina M.**, **Metze F.**, **Hovy E. H.** and **Black A. W.** (2021). NoiseQA: Challenge set evaluation for user-centric question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, Online Event, pp. 2976–2992.
Rayner K., **White S.**, **Johnson R.** and **Liversedge S.** (2006). Raeding wrods with jubmled lettres: There is a cost. *Psychological Science* **17**(3) 192–193.
Renuka D. K. and **Hamsapriya T.** (2010). Email classification for spam detection using word stemming. *International Journal of Computer Applications* **1**(5), 45–47.
Riabi A., **Sagot B.** and **Seddah D.** (2021). Can character-based language models improve downstream task performances in low-resource and noisy language scenarios? In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, Online. Association for Computational Linguistics, pp. 423–436.
Richards J. and **Schmidt R.** (2013). *Longman Dictionary of Language Teaching and Applied Linguistics*. London, UK: Taylor and Francis.
Riordan B., **Flor M.** and **Pugh R.** (2019). How to account for mispellings: Quantifying the benefit of character representations in neural content scoring models. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2019*, Florence, Italy: Association for Computational Linguistics, pp. 116–126.
Röttger P., **Vidgen B.**, **Nguyen D.**, **Waseem Z.**, **Margetts H. Z.** and **Pierrehumbert J. B.** (2021). Hatecheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event*, Bali, ID: Nusa Dua, Association for Computational Linguistics, pp. 41–58.
Rust P., **Lotz J. F.**, **Bugliarello E.**, **Salesky E.**, **de Lhoneux M.** and **Elliott D.** (2023). Language modelling with pixels. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 223)*, Kigali, RW, pp. 1–32.
Sakaguchi K., **Duh K.**, **Post M.** and **Durme B. V.** (2017). Robsut wrod reocginiton via semi-character recurrent neural network. In *Proceedings of the 31st Conference on Artificial Intelligence (AAAI 2017)*, San Francisco, US, pp. 3281–3287.
Salesky E., **Etter D.** and **Post M.** (2021). Robust open-vocabulary translation from visual text representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, Punta Cana, DO, pp. 7235–7252.
Sang E. F. and **De Meulder F.** (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning (CoNLL 2003)*, Edmonton, CA, pp. 142–147.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Sankar C., **Ravi S.** and **Kozareva Z.** (2021). On-device text representations robust to misspellings via projections. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, Online Event, pp. 2871–2876.

Satheesh S., **Beckh K.**, **Klug K.**, **Allende-Cid H.**, **Houben S.** and **Hassan T.** (2025). Robustness evaluation of the German extractive question answering task. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025*, Abu Dhabi, UAE: Association for Computational Linguistics, pp. 1785–1801.

Schulz S., **Pauw G. D.**, **Clercq O. D.**, **Desmet B.**, **Hoste V.**, **Daelemans W.** and **Macken L.** (2016). Multimodular text normalization of dutch user-generated content. *ACM Transactions on Intelligent Systems and Technology* **7**(4), 22.

Sebastiani F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys* **34**(1), 1–47.

Šebesta K., **Bedřichová Z.**, **Štindlová B.**, **Hrdlička M.**, **Hrdličková T.**, **Hana J.**, **Rosen A.**, **Petkevič V.**, **Jelínek T.**, **Škodová S.**, **Janeš P.**, **Lundáková K.**, **Skoumalová H.**, **Šťastný K.** and **Sládek Š.** (2017). CzeSL grammatical error correction dataset (CzeSL-GEC). In *LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL)*. Prague, CZ: Faculty of Mathematics and Physics, Charles University, pp. 452–467.

Shishibori M., **Lee S. S.**, **Oono M.** and **Aoe J.** (2002). Improvement of the LR parsing table and its application to grammatical error correction. *Information Sciences* **148**(1–4), 11–26.

Shook A., **Chabal S.**, **Bartolotti J.** and **Marian V.** (2012). CLEARPOND: Cross-linguistic easy-access resource for phonological and orthographic neighborhood densities. *PLOS ONE* **7**(8), e43230.

Sidiropoulos G. and **Kanoulas E.** (2022). Analysing the robustness of dual encoders for dense retrieval against misspellings. In *Proceedings of the 45th International Conference on Research and Development in Information Retrieval (SIGIR 2022)*, Madrid, ES, pp. 2132–2136.

Specia L., **Li Z.**, **Pino J. M.**, **Chaudhary V.**, **Guzmán F.**, **Neubig G.**, **Durrani N.**, **Belinkov Y.**, **Koehn P.**, **Sajjad H.**, **Michel P.** and **Li X.** (2020). Findings of the WMT 2020 shared task on machine translation robustness. In *Proceedings of the 5th Conference on Machine Translation (WMT 2020)*, Online Event, pp. 76–91.

Sperduti G., **Moreo A.** and **Sebastiani F.** (2021). Garbled-word embeddings for jumbled text. In *Proceedings of the 11th Italian Information Retrieval Workshop 2021 (IIR 2021)*, Bari, IT, pp. 1–6.

Stamatatos E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science & Technology* **60**(3), 538–556.

Subramaniam L. V., **Roy S.**, **Faruquie T. A.** and **Negi S.** (2009). A survey of types of text noise and techniques to handle noisy text. In *Proceedings of the 3th Workshop on Analytics for Noisy Unstructured Text Data (AND 2009)*, Barcelona, ES, pp. 115–122.

Touvron H., **Lavril T.**, **Izacard G.**, **Martinet X.**, **Lachaux M.**, **Lacroix T.**, **Rozière B.**, **Goyal N.**, **Hambro E.**, **Azhar F.**, **Rodriguez A.**, **Joulin A.**, **Grave E.** and **Lample G.** (2023). Llama: open and efficient foundation language models, arXiv: 2302.13971 [cs.CL].

Vaibhav Singh S and **Neubig G.** (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 (NAACL-HLT 2019)*, Minneapolis, US, pp. 1916–1920.

van der Goot R., **Ramponi A.**, **Caselli T.**, **Cafagna M.** and **Mattei L. D.** (2020). Norm it! lexical normalization for italian and its downstream effects for dependency parsing. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020*, Marseille, France: European Language Resources Association, pp. 6272–6278.

van der Goot R., **van Noord R.** and **van Noord G.** (2018). A taxonomy for in-depth evaluation of normalization for user-generated content. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, JP, pp. 684–688.

van Rijsbergen C.J. (1979). *Information Retrieval*. 2nd ed. London, UK: Butterworth-Heinemann.

Vinciarelli A. (2005). Noisy text categorization. *IEEE Transactions on Pattern and Machine Intelligence* **27**(12), 1882–1895.

Wang B., **Chen W.**, **Pei H.**, **Xie C.**, **Kang M.**, **Zhang C.**, **Xu C.**, **Xiong Z.**, **Dutta R.**, **Schaeffer R.**, **Truong S. T.**, **Arora S.**, **Mazeika M.**, **Hendrycks D.**, **Lin Z.**, **Cheng Y.**, **Koyejo S.**, **Song D.** and **Li B.** (2023). DecodingTrust: A comprehensive assessment of trustworthiness in GPT models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems (NeurIPS 2023)*, New Orleans, New Orleans, LA, USA, pp. 1–84.

Wang B., **Xu C.**, **Wang S.**, **Gan Z.**, **Cheng Y.**, **Gao J.**, **Awadallah A. H.** and **Li B.** (2021a). Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, virtual*, pp. 1–13.

Wang C., **Gu T.**, **Wei Z.**, **Gao L.**, **Song Z.** and **Chen X.** (2025). Word form matters: Llms' semantic reconstruction under typoglycemia. In Che W., Nabende J., Shutova E. and Pilehvar M. T. (eds), *Findings of the Association for Computational Linguistic , ACL 2025*. Vienna, Austria: Association for Computational Linguistics, pp. 16870–16885.

Wang H., **Zhang P.** and **Xing E.** P. (2020). Word shape matters: Robust machine translation with visual embedding. arXiv: 2010.09997 [cs.CL].

Wang J., **Hu X.**, **Hou W.**, **Chen H.**, **Zheng R.**, **Wang Y.**, **Yang L.**, **Ye W.**, **Huang H.**, **Geng X.**, **Jiao B.**, **Zhang Y.** and **Xie X.** (2024). On the robustness of ChatGPT: An adversarial and out-of-distribution perspective. *IEEE Computer Society Technical Committee on Data Engineering Bulletin* **47**(1), 48–62.

https://doi.org/10.1017/nlp.2026.10020 Published online by Cambridge University Press

Wang Y., **Wang Y.**, **Dang K.**, **Liu J.** and **Liu Z.** (2021b). A comprehensive survey of grammatical error correction. *ACM Transactions on Intelligent Systems and Technology (TIST)* **12**(5), 1–51.

Wu T., **Wen S.**, **Xiang Y.** and **Zhou W.** (2018). Twitter spam detection: Survey of new approaches and comparative study. *Computers & Security* **76**, 265–284.

Yang R. and **Gao Z.** (2019). Can machines read jmulbed senetcnes? Unpublished manuscript. *Available at:* https://runzhe-yang.science/demo/jumbled.pdf.

Zellers R., **Holtzman A.**, **Bisk Y.**, **Farhadi A.** and **Choi Y.** (2019). HellaSwag: Can a machine really finish your sentence?. In Korhonen A., Traum D. and Màrquez L. (eds), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4791–4800.

Zesch T. (2012). Measuring contextual fitness using error contexts extracted from the Wikipedia revision history. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, Avignon, FR, pp. 529–538.

Zhang Z., **Hao B.**, **Li J.**, **Zhang Z.** and **Zhao D.** (2024). E-bench: Towards evaluating the ease-of-use of large language models. arXiv: 2105.09825 [cs.CL].

Zheng R., **Liu H.**, **Ma M.**, **Zheng B.** and **Huang L.** (2019). Robust machine translation with domain sensitive pseudo-sources: Baidu-OSU WMT19 MT robustness shared task system report. In *Proceedings of the 4th Conference on Machine Translation (WMT 2019)*, Firenze, IT, pp. 559–564.

Zhou J. T., **Zhang H.**, **Jin D.**, **Peng X.**, **Xiao Y.** and **Cao Z.** (2020). RoSeq: Robust sequence labeling. *IEEE Transactions on Neural Networks and Learning Systems* **31**(7), 2304–2314.

Zhou S., **Zeng X.**, **Zhou Y.**, **Anastasopoulos A.** and **Neubig G.** (2019). Improving robustness of neural machine translation with multi-task learning. In *Proceedings of the 4th Conference on Machine Translation (WMT 2019)*, Firenze, IT, pp. 565–571.

Zhu K., **Wang J.**, **Zhou J.**, **Wang Z.**, **Chen H.**, **Wang Y.**, **Yang L.**, **Ye W.**, **Zhang Y.**, **Gong N.**, *et al.* (2023). PromptRobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, pp. 57–68.

Cite this article: Sperduti G and Moreo A. Misspellings in natural language processing: A survey of recent literature. *Natural Language Processing* https://doi.org/10.1017/nlp.2026.10020