electronics

MDPI

Review

Survey of Automatic Spelling Correction

Daniel Hládek *, Ján Staš and Matúš Pleva

Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Němcovej 32, 040 01 Košice, Slovakia; jan.stas@tuke.sk (J.S.); matus.pleva@tuke.sk (M.P.)

* Correspondence: daniel.hladek@tuke.sk; Tel.: +421-055-602-2298

Received: 13 August 2020; Accepted: 6 October 2020; Published: 13 October 2020

Abstract: Automatic spelling correction has been receiving sustained research attention. Although each article contains a brief introduction to the topic, there is a lack of work that would summarize the theoretical framework and provide an overview of the approaches developed so far. Our survey selected papers about spelling correction indexed in Scopus and Web of Science from 1991 to 2019. The first group uses a set of rules designed in advance. The second group uses an additional model of context. The third group of automatic spelling correction systems in the survey can adapt its model to the given problem. The summary tables show the application area, language, string metrics, and context model for each system. The survey describes selected approaches in a common theoretical framework based on Shannon's noisy channel. A separate section describes evaluation methods and benchmarks.

Keywords: spelling correction; natural language processing; diacritization; error model; context model

1. Introduction

There are many possible ways to write the same thing. Written text sometimes looks different from what the reader or the author expects. Creating apprehensive and clear text is not a matter of course, especially for people with a different mother language. An unusually written word in a sentence makes a spelling error.

A spelling error makes the text harder to read and, worse, harder to process. Natural language processing requires normalized forms of a word because incorrect spelling or digitization of text decreases informational value. A spelling error, for example, in a database of medical records, diminishes efficiency of the diagnosis process, and incorrectly digitized archive documents can influence research or organizational processes.

A writer might not have enough time or ability to correct spelling errors. Automatic spelling correction (ASC) systems help to find the intended form of a word. They identify problematic words and propose a set of replacement candidates. The candidates are usually sorted according to their expected fitness with the spelling error and the surrounding context. The best correction can be selected interactively or automatically.

Interactive spelling correction systems underline incorrectly written words and suggest corrections. A user of the system selects the most suitable correction. This scenario is common in computer-assisted proofreading that helps with the identification and correction of spelling errors. Interactive spelling correction systems improve the productivity of professionals working with texts, increase convenience when using mobile devices, or correct Internet search queries. They support learning a language, text input in mobile devices, and web search engines. Also, interactive spelling correction systems are a component of text editors and office systems, optical character recognition (OCR) systems, and databases of scanned texts.

Most current search engines can detect misspelled search queries. The suggestion is shown interactively for each given string prefix. A recent work by Cai and de Rijke [1] reviewed approaches for correcting search queries.

A large quantity of text in databases brought new challenges. An automatic spelling correction system can be a part of a natural language processing system. Text in the database has to be automatically corrected because interactive correction would be too expensive. The spelling correction system automatically selects a correction candidate according to the previous and following texts. Noninteractive text normalization can improve the performance of information retrieval or semantic analysis of a text.

Figure 1 displays the process of correction-candidate generation and correction. The error and context models contribute to ranking of the candidate words. The result of automatic correction is a sequence of correction candidates with the best ranking.

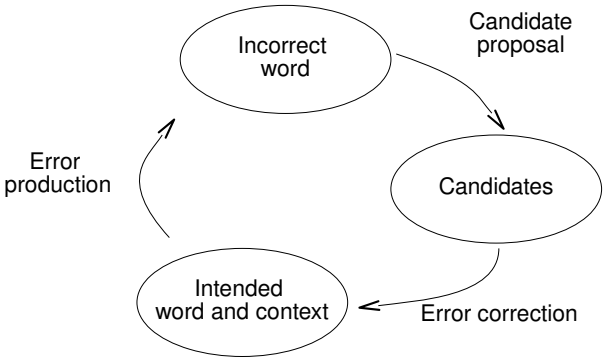

Figure 1. Interactive processes of error production and correction.

In the next section, you'll find an explanation of the method we used to select and sort the articles in this report. Subsequently, in Section 3, we describe the characteristic spelling errors and divide them into groups according to how they originated. Section 4 defines the task of correcting spelling errors and describes the ASC system. This survey divides the ASC systems into three groups, each with its section: a priori spelling correction (Section 5), spelling correction in the context (Section 6), and spelling correction with a learning error model (Section 7). Section 8 introduces the methods of evaluation and benchmarking. The concluding Section 9 summarizes the survey and outlines trends in the research.

2. Methodology

The survey selected papers about spelling correction indexed in Scopus (http://scopus.com) and Web of Science (https://apps.webofknowledge.com) (WoS) from 1991 to 2019. It reviews the state-of-the-art and maps the history from the previous comprehensive survey provided by Kukich [2] in 1992.

First, we searched the indices with a search query "spelling correction" for the years 1991–2019. Scopus returned 1315 results, WoS returned 794 results. We excluded 149 errata, 779 corrections, 7 editorials, 45 reviews, and around 140 papers without any citations from both collections. We removed 250 duplicates, and we received 740 results (440 journal articles and 300 conference papers). We read the titles and abstracts of the remaining papers and removed 386 works that are not relevant to the topic of automatic spelling correction.

We examined the remaining 354 documents. Then, we removed articles without clear scientific contribution to spelling correction, without proper evaluation, or that just repeated already known things. We examined, sorted, and put the remaining 119 items into tables. We included additional references that explain essential theoretical concepts and survey papers about particular topics in the surrounding text.

First, we defined the spelling correction problem and established a common theoretical framework. We described the three main components of a spelling correction system.

This work divides the selected papers into three groups. The first group uses a set of expert rules to correct a spelling error. The second group adds a context model to rearrange the correction candidates with the context. The third group learns error patterns from a training corpus.

Each group of methods has its own section with a summarizing table. The main part of the survey is the summary tables. The tables briefly describe the application area, language, error model, and context model of the spelling correction systems. The tables are accompanied by a description of the selected approaches.

The rows in the tables are sorted chronologically and according to author. We selected chronological order because it shows the general scientific progress in spelling correction in the particular components of the spelling correction system. An additional reference in the table indicates if one approach enhances the previous one.

Special attention is paid to the evaluation methods. This section identifies the most frequent evaluation methods, benchmarks and corpora.

3. Spelling Errors

The design of an automatic spelling correction system requires knowledge of the creation process of a spelling error [3]. There are several works about spelling errors. A book by Mitton [4] analyzed spelling-error types and described approaches to construct an automatic spelling correction system. The authors in Yannakoudakis and Fawthrop [5] demonstrated that the clear majority of spelling errors follow specific rules on the basis of phonological and sequential considerations. The paper [5] introduced and described three categories of spelling errors (consonantal, vowel, and sequential) and presented the analysis results of 1377 spelling error forms.

Moreover, the authors in Kukich [2], Toutanova and Moore [6], and Pirinen and Lindén [7] divided spelling errors into two categories according to their cause:

1. Cognitive errors (also called orthographic or consistent): They are caused by the disabilities of the person that writes the text. The correct way of writing may be unknown to the writer. The writer could have dyslexia, dysgraphia, or other cognitive problems. The person writing the text could just be learning the language and not know the correct spelling. This set of errors is language- and user-specific because it is more dependent on using the rules of the language [7].
2. Typographic errors (also called conventional): They are usually related to technical restrictions of the input device (physical or virtual keyboard, or OCR system) or depend on the conditions of the environment. Typing in haste often causes substitution of two close keys. Typographic errors caused by hasty typing are usually language-agnostic (unrelated to the language of the writer), although they can depend on local keyboard mapping or a localized OCR system [7].

Examples of typographic and cognitive spelling errors are in Table 1.

Table 1. Examples of cognitive and typographic errors.

Error Type	Example
Cognitive error:	I don't know the correct spelling of Levenstain distance.
Typographic:	THis sentence was typed in haser.
Typographic (OCR):	SUppLEMENTAhy lNFOhMATlON.
Typographic (Diacritic):	The authors of this article are Daniel Hladek, Matus Pleva and Jan Stas.

Note: The spelling errors are underlined.

OCR errors are a particular type of typographic error caused by software. The process of document digitization and optical character recognition often omits or replaces some letters in a typical way. Spelling correction is part of postprocessing of the digitized document because OCR systems are

usually proprietary and difficult to adapt. Typical error patterns appear in OCR texts [8]. The standard set for evaluation of an OCR spelling correction system is the TREC-5 Confusion Track [9].

Some writing systems (such as Arabic, Vietnamese, or Slovak) use different character variants that change the meaning of the word. The authors in [10] confirmed that the omission of diacritics is a common type of spelling error in Brazilian Portuguese. Texts in Modern Standard Arabic are typically written without diacritical markings [11]. This is a typographic error when the author omits additional character markings and expects the reader to guess the original meaning. The missing marks usually present short vowels or modification of the letter. They are placed either above or below the graphemes. The process of adding vowels and other diacritic marks to Arabic text can be called diacritization or vowelization [11]. Azmi and Almajed [12] focused on the problem of Arabic diacritization (adding missing diacritical markings to Arabic letters) and proposed an evaluation metric, and Asahiah et al. [13] published a survey of Arabic diacritization techniques.

4. Automatic Spelling Correction

An automatic spelling correction system detects a spelling error and proposes a set of candidates for correction (see Figure 2). Kukich [2] and Pirinen and Lindén [7] divide the whole process into three steps:

1. detection of an error;
2. generation of correction candidates;
3. ranking of candidate corrections.

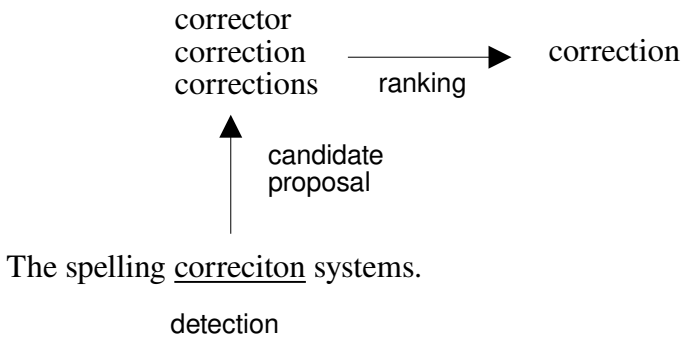

Figure 2. Process of automatic spelling correction.

4.1. Error Detection

A word could either be new or just uncommon, could be a less-known proper name, or could belong to another language. However, a correctly spelled word could be semantically incorrect in a sentence. Kukich [2] divided spelling errors according to the dictionary of correct words:

- real-word errors, where the word is spelled incorrectly but its form is in the dictionary of correct words, and
- non-word errors, where the incorrect word form is not in the dictionary of correct words.

Most spelling correction systems detect a non-word error by searching for it in a dictionary of correct words. This step requires a fast-lookup method such as hash table [14] or search tree [15,16].

Many non-word error spelling correction systems use open-source a priori spelling systems, such as Aspell or Hunspell for error detection, correction-candidate generation, and preliminary candidate ranking.

An automatic spelling correction system identifies real-word errors by semantic analysis of the surrounding context. More complex error-detection systems may be used to detect words that are correctly spelled but do not fit into the syntactic or semantic context. Pirinen and Lindén [7] called it real-word error detection in context.

Real-word errors are hard to detect because detection requires semantic analysis of the context. The authors in [17] used a language model to detect and correct a homophonic real-word error in the Bangla language. The language model identifies words that are improbable with the current context.

Boytsov [18] examined methods for indexing a dictionary with approximate matching. Deorowicz and Ciura [19] claim that a lexicon of all correct words could be too large. Too large a lexicon can lead to many real-word errors or misdetection of obscure spellings.

The situation is different for languages where words are not separated by spaces (for example, Chinese). The authors in [20] transformed characters into a fixed-dimensional word-vector space and detected spelling errors by conditional random field classification.

4.2. Candidate Generation

ASC systems usually select correction candidates from a dictionary of correct words after detection of a spelling error. Although it is possible to select all correct words as correction candidates, it is reasonable to restrict the search space and to inspect only words that are similar to the identified spelling error.

Zhang and Zhang [21] stated that the task of similarity joining is to find all pairs of strings for which similarities are above a predetermined threshold, where the similarity of two strings is measured by a specific distance function. Kernighan et al. [22] proposed a simplification to restrict the candidate list to words that differ with just one edit operation of the Damerau–Levenshtein edit distance—substitution, insertion, deletion, or replacement of succeeding letters [23].

The spelling dictionary generates correction candidates for the incorrect word by approximately searching for similar words. The authors in [24] used a character-level language model trained on a dictionary of correct words to generate a candidate list. Reffle [25] used a Levenshtein automaton to propose the correction candidates. Methods of approximate searching were outlined in a survey published by Yu et al. [26].

An index often speeds up an approximate search in the dictionary. The authors in [19,27] converted the lexicon into a finite-state automaton to speed up searching for a similar string.

4.3. Ranking Correction Candidates

A noisy-channel model proposed by Shannon [28] described the probabilistic process of producing an error. The noisy channel transfers and distorts words (Figure 3).

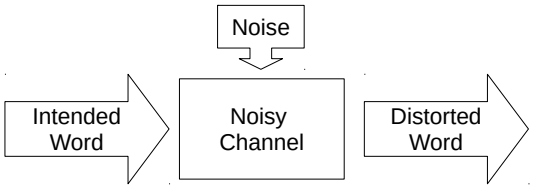

Figure 3. Word distorted by noisy channel.

The noisy-channel model expresses similarity between two strings as a probability of transforming one string into another. Probability $P(s|w)$ that a string s is produced instead of word w describes how similar the two strings are. The similarity between two strings is defined by an expert or depends on a training corpus with error patterns.

A more formal definition of automatic spelling correction uses the maximum-likelihood principle. Brill and Moore [29] defined the automatic spelling correction of a possibly incorrect word s as finding the best correction candidate w_b from a list of possible correction candidates $w_i \in W$ with the highest un-normalized probability:

$$w_b = \arg \max_{w_i \in C(s)} P(s|w_i)P(w_i) \,, \tag{1}$$

where $P(s|w_i)$ is the probability of producing string s instead of word w_i and $P(w_i)$ is the probability of producing word w_i. $C(s)$ is a function that returns valid words from dictionary W that serve as correction candidates for erroneous string s.

4.4. Components of Automatic Spelling Correction Systems

Equation (1) by Brill and Moore [29] identified three components of an automatic spelling correction system. The components are depicted in Figure 4:

1. Dictionary: It detects spelling errors and proposes correction candidates $w_i \in C$ for each input token. $C(s)$ is a list of correction candidates w_i for a given token s. The list of correction candidates belongs to the set of all correct words ($C(s) \in W$). If the dictionary does not propose any candidate, the word is considered correct.
2. Error model (channel model) $P(s|w_i)$: It is an essential component of the automatic spelling correction system. It measures the "fitness" of the correction candidate with the corrected string. The model expresses the similarity of strings w_i and s or the probability of producing string s instead string w_i. This measure does not have to be purely probabilistic but can be similar to a distance between the two strings. The non-probabilistic string distance can always be converted into probabilistic string similarity (see Equation (3) in Section 6). An error model allows for identification of the most probable errors and consequently the most probable original forms.
3. Context model (source model $P(w_i)$, the prior model of word probabilities): This expresses the probability of correct word occurrence and often takes the context of the word into account. Candidates that best fit into the current context have a higher probability of being the intended word. The context model focuses on finding the best correction candidate by using the context of the incorrect word and statistical methods of classification. The model observes features that are outside the inspected word and improves the evaluation of candidate words. It can detect an unusual sequence of features and identify real-word errors.

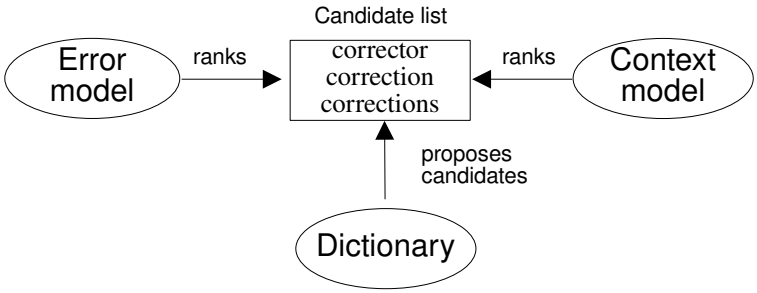

Figure 4. Components of an automatic spelling correction system.

5. Spelling Correction with a Priori Error Model

A combination of error and context models is often not necessary. In some scenarios, a set of predefined transcription rules can correct a spelling error. An expert identifies characteristic string transcriptions. These rules are given in advance (a priori) by someone who understands the problem.

Approaches in this group detect non-word errors and propose a list of correction candidates that are similar to the original word (presented in Table 2). The a priori error model works as a guide in the search for the best-matching original word; best-matching words are proposed first, and it is easy to select the correction.

A schematic diagram for an ASC system with a priori error model is in Figure 5. The input of the a priori error model is an erroneous word. The spelling system applies one or several transcription operations to the spelling error to create a correction candidate. The rank of the correction candidate depends on the weights of the transcription rules. The output of the a priori error model is a sorted list with correction candidates.

Table 2. Summary of a priori spelling correction systems.

Reference	Application	Language	Error Model
Khairul Islam et al. [30], 2019	General	Bangla	LD
Hawezi et al. [31], 2019	General	Kurdish	LD, DLD, LCS
Thaiprayoon et al. [32], 2018	Search query	Thai	LD, Soundex
Christanti et al. [33], 2018	General	Indonesian	DLD
Hagen et al. [34], 2017	Search query	English	DLD
Sakuntharaj and Mahesan [35], 2016	General	Tamil	LD, common *n*-grams
Vobl et al. [36], 2014	OCR, historical	Old German	Interactive
Rees [37], 2014	Animal taxonomy	Latin	Soundex
Mühlberger et al. [38], 2014	OCR, historical	German	Interactive
Patrick and Nguyen [39], 2014	General, medical	English	Interactive
Kashefi et al. [40], 2013	Diacritization	Farsi	Modified DLD
Andrade et al. [41], 2012	General	Portuguese	DLD
Sha et al. [42], 2011	General	Chinese	Keyboard-based edit distance
Reffle [25], 2011	OCR, historical	Old German	LD, FSA
Naji and Savoy [43], 2011	General, historical	Middle High German	Stemmer
Bustamante et al. [44], 2006	General	Spanish	Interactive + generalized LD
Deorowicz and Ciura [19], 2005	General	English	FSA
UzZaman and Khan [45], 2005	General	Bangla	Bangla double metaphone
Vilares et al. [27], 2004	General	Galician	FSA
van Delden et al. [46], 2004	General	English	LD, stemming
Schulz and Mihov [47], 2002	General	Bulgarian, German	FSA
Taghva and Stofsky [48], 2001	OCR	English	Interactive + LCS subsequence
Vagelatos et al. [49], 1995	General	Greek	Interactive

Note: DLD, Damerau–Levenshtein distance; FSA, finite-state automaton; LCS, longest common subsequence; LD, Levenshtein distance; OCR, optical character recognition.

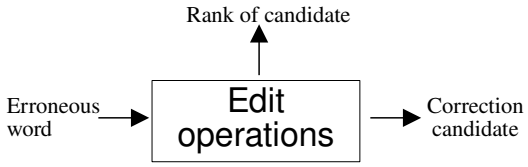

Figure 5. A priori spelling correction.

The most commonly used open-source spelling systems are Aspell (http://aspell.net) and Hunspell (http://hunspell.github.io/). Hunspell is a variant of Aspell with a less restrictive license, used in LibreOffice word processor, Firefox web browser, and other programs. They are available as a standalone text filter or as a compiled component in other spelling systems or programs. The basic component of the Aspell system is a dictionary of correct words, available for many languages. The dictionary file contains valid morphological units for the given language (prefixes, suffixes, or stems). The dictionary is compiled into a state machine to speed up searching for correction candidate words.

Aspell searches for sounds-like equivalents (computed for English words by using the Metaphone algorithm) up to a given edit distance (the Damerau–Levenshtein distance) [50]. The detailed operation of the spelling correction of Aspell is described in the manual (http://aspell.net/man-html/Aspell-Suggestion-Strategy.html#Aspell-Suggestion-Strategy).

5.1. Edit Distance

Edit distance expresses the difference between two strings as a nonnegative real number by counting edit operations that are required to transform one string into another. The two most commonly used edit distances are the Levenshtein edit distance [51] and the Damerau–Levenshtein distance [52]. Levenshtein identifies atomic edit operations such as

- Substitution: replaces one symbol into another;
- Deletion: removes a symbol (or replaces it with an empty string ϵ); and
- Insertion: adds a symbol or replaces an empty string ϵ with a symbol.

In addition, the Damerau–Levenshtein distance adds the operation of

- Transposition, which exchanges two subsequent symbols.

The significant difference between the Levenshtein distance (LD) and the Damerau–Levenshtein distance (DLD) is that the Levenshtein distance does not consider letter transposition. The edit operation set proposed by Levenshtein [51] did not consider transposition as an edit operation because the transposition of two subsequent letters can be substituted by deletion and insertion or by two substitutions. The Levenshtein distance allows for representation of the weights of edit operations by a single letter-confusion matrix, which is not possible for DLD distance.

Another variation of edit distance is longest common subsequence (LCS) [53]. It considers only insertion and deletion edit operations. The authors in [54] proposed an algorithm for searching for the longest common sub-string with the given number of permitted mismatches. More information about longest-common-subsequence algorithms can be found in a survey [55].

5.2. Phonetic Algorithms

Many languages have difficult rules for pronunciation and writing, and it is very easy to make a spelling mistake if rules for writing a certain word are not familiar to the writer. A word is often replaced with a similarly sounding equivalent with a different spelling.

An edit operation in the phonetic algorithm describes how words are pronounced. They recursively replace phonetically important parts of a string into a special representation. If the phonetic representation of two strings is equal, the strings are considered equal. In other words, a phonetic algorithm is a binary relation of two strings that tells whether two strings are pronounced in a similar way:

$$D(s_s, s_t) \rightarrow 0 \text{ or } 1 \, . \tag{2}$$

The phonetic algorithm is able to identify a group of phonetically similar words to some given string (e.g., to some unknown proper noun). It helps to identify names that are pronounced in a similar way or to discover the original spelling of an incorrectly spelled word. Two strings are phonetically similar only if their phonetic forms are equal.

Phonetic algorithms for spelling corrections and record linkage are different from phonetic algorithms used for speech recognition because they return just an approximation of the true phonetic representation.

One of the first phonetic algorithms is Soundex (U.S. Patent US1435663). Its original purpose was the identification of similar names for the U.S. Census. The algorithm transforms a surname or name so that names with a similar pronunciation have the same representation. It allows for the identification of similar or possibly the same names. The most phonetically important letters are consonants. Most vowels are dropped (except for in the beginning), and similar consonants are transformed into the same representation. Other phonetic algorithms are Shapex [56] and Metaphone [57]. Evaluation of several phonetic-similarity algorithms on the task of cognate identification was done by Kondrak and Sherif [58].

6. Spelling Correction in Context

An a priori model is often not sufficient to find out the best correction because it takes only incorrect word into account. The spelling system would perform better if it could distinguish whether the proposed word fits with its context. It is hard to decide which correction is more useful if we do not know the surrounding sentence. For example, if a correction for string “smilly” is “smelly”, the correction “smiley” can be more suitable for some contexts.

Approaches in this group are summarized in Tables 3 and 4. The components and their functions are displayed in Figure 4. The authors in [59] described multiple methods of correction with context. This group of automatic spelling correction systems use a probabilistic framework by Brill and Moore [29] defined in the Equation (1). The error models in this group usually use the a priori rules (edit distance and phonetic algorithms). The context model is usually an n-gram language model. Some approaches noted below use a combination of multiple statistical models.

Table 3. Spelling correction systems with learning of context model—part I.

Reference	Application	Language	Context Model	Error Model
Azmi et al. [60], 2019	General, OCR	Arabic	LM	LD, DLD
Dong et al. [61], 2019	MT	Uygur, Chinese	LM, BLEU score	LD
Yazdani et al. [62], 2019	Medical	Farsi	LM	DLD
Damnati et al. [63], 2018	POS	French	Word embedding	DLD
Dashti [64], 2018	General	English	LM	CFG
Fahda and Purwarianti [65], 2018	General	Indonesian	LM, POS, Viterbi	DLD
Heyman et al. [66], 2018	General	Dutch	Suffix probability	BiLSTM
Mashod Rana et al. [17], 2018	General	Bangla	Golding and Schabes [67]	WCS
Dziadek et al. [68], 2017	Medical ontology	Swedish	LM, POS	LD
Sorokin [69], 2017	General	Russian	LM, LR	LD, Metaphone
Zhao et al. [70], 2017	General	Chinese	CRF, decoder	Graph
de Mendonça Almeida et al. [71], 2016	General	Brazilian Portuguese	Decision tree	Modified Soundex
Lv et al. [72], 2016	OCR, Medical	Chinese	LM, ME	WCS
Melero et al. [73], 2016	General	Spanish	LM	WCS
Mirzababaei and Faili [74], 2016	General	Farsi, English	LM, SVM, PMI	DLD
Sorokin and Shavrina [75], 2016	General	Russian	LM, LR	LD
Vilares et al. [76], 2016	IR	Cross-language	POS	Character *n*-grams, DLD
Lhoussain et al. [77], 2015	General	Arabic	LM	LD
Ferrero et al. [78], 2014	General, proofreading	Spanish	Bayes	Interactive
Miangah [14], 2014	General	Farsi	Word frequency	Letter *n*-grams, DLD
Pirinen and Lindén [7], 2014	General	Finish, Greenlandic	WFST, LM	WFST
Sagiadinos et al. [79], 2014	General	Greek	Id3, C4.5, k-NN, naïve Bayes, RF	Suffix

Note: BLEU, bilingual evaluation understudy; BiLSTM, bidirectional long short-term memory; CFG, context-free grammar; CRF, conditional random fields; IR, information retrieval; k-NN, k-nearest neighbors; LM, language model; LR, linear regression; ME, maximum entropy; POS, part-of-speech tagging; PMI, pointwise mutual information; RF, random forests; SVM, support vector machine; WCS, word-confusion set; WFST, weighted finite-state transducer.

Table 4. spelling correction systems with learning of context model—part II.

Reference	Application	Language	Context Model	Error Model
Ehsan and Faili [80], 2013	General	Farsi, English	SMT, ME	DLD
Hladek et al. [81], 2013	General	Slovak	LM, HMM	Aspell
Flor [59], 2012	General, proofreading	English	LM, Bouma [82]	Custom
Alkanhal et al. [83], 2012	General	Arabic	A-star, LM	DLD
Grozea [84], 2012	Diacritic	Romanian	LM, HMM	Trivial WCS
Stüker et al. [85], 2011	General, diagnosis	German	HMM, LM	Phonetic algorithm
Wong and Glance [86], 2011	General, medical	English	Bayes	Aspell
Abdulkader and Casey [87], 2009	OCR	English	ANN	Interactive
Ahmed et al. [50], 2009	Search query	English	Ternary search trees, letter *n*-grams	
Farooq et al. [88], 2009	Handwritten OCR	English	Topic LM, ME	Trivial WCS
Carlson and Fette [89], 2007	General	English	Banko and Brill [90]	Aspell
Mykowiecka and Marciniak [91], 2006	General, medical	Polish	LM	Modified LD
Héja and Surján [92], 2003	General, medical	Hungarian	*n*-gram tree	Interactive
Jin et al. [93], 2003	OCR	English	ME	WCS
Ruch et al. [94], 2003	General, medical	English, French	POS, ME, WSD	Interactive
Li and Wang [95], 2002	General	Chinese	Golding and Roth [96]	LD
Banko and Brill [90], 2001	General	English	Bayes classifier ensemble	WCS
Carlson et al. [97], 2001	General	English	Golding and Roth [96]	WCS
Ruch et al. [98], 2001	General, medical	French	POS, WSD	Interactive
Golding and Roth [96], 1999	General	English	Winnow algorithm	WCS
Jones and Martin [99], 1997	General	English	LSA	WCS
Golding and Schabes [67], 1996	General	English	Naïve Bayes	WCS

Note: ANN, artificial neural network; HMM, hidden Markov model; LM, language model; LSA, latent semantic analysis; ME, maximum entropy; OCR, optical character recognition; POS, part-of-speech; SMT, statistical machine translation; WCS, word-confusion set; WSD, word-sense disambiguation.

The edit distance $D(s|w)$ of the incorrect word s and a correction candidate w in the a priori error model is a positive real number. In order to fit the probabilistic framework, it can be converted into the probabilistic framework by taking a negative logarithm [100]:

$$P(s|w_i) = -\log D(s, w) \, . \tag{3}$$

Methods of spelling correction in context are similar to morphological analysis, and it is possible to use similar methods of disambiguation from part-of-speech taggers in a context model of automatic spelling correction systems.

6.1. Language Model

The most common form of a language model is n-gram language model, calculated from the frequency of word sequences of size n. It gives the probability $P(w_i|w_{i-1,i-(n-1)})$ of a candidate word given its history of $(n-1)$ words. If the given n-gram sequence is not presented in the training corpus, the probability is calculated by a back-off that considers shorter contexts. The n-gram language model only depends on previous words, but other classifiers can make use of arbitrary features in any part of the context. The language model is usually trained on a training corpus that represents language with correct spelling.

Neural language modeling brought new possibilities, as it can predict a word given arbitrary surrounding context. A neural network maps a word into a fixed-size embedding vector. Embedding vectors form a semantic space of words. Words that are close in the embedding space usually occur in the same context and are thus semantically close. This feature can be used in a spelling correction system to propose and rank a list of correction candidates [63,101,102].

6.2. Combination of Multiple Context Models

Context modeling often benefits from a combination of multiple statistical models. A spelling system proposed by Melero et al. [73] used a linear combination of language models, each with a certain weight. Each language model can focus on a different feature: lowercase words, uppercase words, part-of-speech tags, and lemmas.

The authors in [67] proposed a context model with multiple Bayesian classifiers. The first component of the context model is called “trigrams". This system uses parts of speech as a feature for classification. The first part of the model assigns the highest probability to a candidate word and its context containing the most probable part-of-speech tags. The second part of the context model is a naïve Bayes classifier that takes the surrounding words and collocations (preceding word and current tag) .

Another form of a statistical classifier for the context modeling with multiple models is the Winnow algorithm [96,103]. This approach uses several Winnow classifiers trained with different parameters. The final rank is their weighted sum.

The model uses the same features (occurrence of a word in context and collocation of tags and surrounding word) as those in the previous approach [67]. The paper by Golding and Roth [96] was followed by Carlson et al. [97], which used a large-scale training corpus. Also, Li and Wang [95] proposed a similar system for Chinese spelling correction.

An approach published by Banko and Brill [90] proposed a voting scheme that utilized four classifiers. This approach focused on learning by using a large amount of data—over 100 million words. It uses a Winnow classifier, naïve Bayes classifier, perceptron, and a simple memory-based learner. Each classifier has a complementarity score defined by Brill et al. [104] and is separately trained. The complementarity score indicates how accurate the classifier is.

6.3. Weighted Finite-State Transducers

If components of an ASC system (dictionary, error model, or context model) can be converted into a state machine, it is possible to create a single state machine by composing individual components. The idea of finite-state spelling was formalized by Pirinen and Lindén [7]. They compared finite-state automatic spelling correction systems with other conventional systems (Aspell and Hunspell) for English, Finnish, and Icelandic on the corpus of Wikipedia edits. They showed that this approach had comparable performance to that of others.

A weighted state transducer (WFST) is a generalization of a finite-state automaton, where each transcription rule has an input string, output string, and weight. One rule of the WFST system represents a single piece of knowledge about spelling correction—an edit operation of the error model or a probability of succeeding words in the context model.

Multiple WFSTs (dictionary, error model, and context model) can be composed into a single WFST by joining their state spaces and by removing useless states and transcription rules. After these three components are composed, the resulting transducer can be searched for the best path, which is the sequence of best-matching letters.

For example, the approach by Perez-Cortes et al. [105] took a set of hypotheses from the OCR. The output from OCR is an identity transducer (an automaton that transcribes the set of strings to the same set of strings) with weights on each transition that represents the probability of a character in the hypothesis. The character-level n-gram model represents a list of valid strings from the lexicon. The third component of the error model is a letter-confusion matrix calculated from the training corpus. The authors in [106,107] used handcrafted Arabic morphological rules to construct a WFST for automatic spelling correction.

A significant portion of text errors involves running together two or more words (e.g., ofthe) or splitting a single word (sp ent, th ebook) [2]. Weighted finite-state transducer (WFST) systems can identify word boundaries if the spacing is incorrect (http://openfst.org/twiki/bin/view/FST/FstExamples). However, inserting or deleting a space is still considered problematic because spaces have the annoying characteristic of not being handled by edit-distance operations [106].

7. Spelling Correction with Learning Error Model

The previous sections presented spelling correction systems with a fixed set of rules, prepared in advance by an expert. This section introduces approaches where the error model learns from a training corpus. The optimization algorithm iteratively updates the parameters of the error model (e.g., weights of the edit operations) to improve the quality of the ASC system.

A diagram in Figure 6 displays a structure of a learning error model. The algorithm for learning the error model uses the expectation-maximization procedure. A complete automatic spelling correction system contains a context model that is usually learned separately. The authors in [108] proposed to utilize the context model in the learning of the error model. Context probability is taken into account during the expectation step. Some approaches do not consider context at all. A comparison of approaches with the learning error model is shown in Tables 5 and 6.

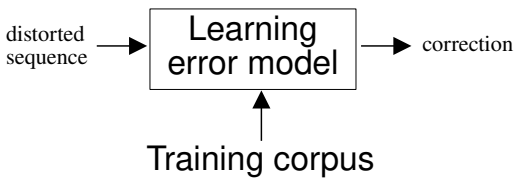

Figure 6. Spelling correction with learning error model

Table 5. Spelling correction systems with learning of context model and error model—part I.

Reference	Application	Language	Context Model	Error Model
Han et al. [109], 2019	General	Chinese	N/a	BiLSTM seq2seq [110]
Jain et al. [111], 2019	General	Hindi	LM	LD, LCM
Kinaci [24], 2019	General	Turkish	N/a	LSTM character LM
Lu et al. [112], 2019	Diacritic	Mongolian	N/a	Evolved transformer seq2seq
Mammadov [113], 2019	General	Azerbaijani	N/a	Seq2seq [110]
Roy [114], 2019	General	English	N/a	Seq2seq transformer
Yang et al. [115], 2019	Speech postprocessing	Chinese	N/a	CRF, seq2seq [110], BERT, character embeddings
Zaky and Romadhony [102], 2019	General	Indonesian	N/a	POS, word embeddings, BiLSTM seq2seq [110]
Zhang et al. [116], 2019	Speech postprocessing	Chinese	N/a	Transformer seq2seq
Zhou et al. [117], 2019	General	English	N/a	BiLSTM seq2seq [110]
Barteld et al. [118], 2018	Historical POS	Middle High German	N/a	Character LM
Sooraj et al. [119], 2018	General	Malayan	N/a	LSTM character LM
Sbattella and Tedesco [120], 2018	General	Italian	N/a	seq2seq LSTM
Fivez et al. [101], 2017	Medical	English, Dutch	Word embedding	DLD, double-Metaphone, character embeddings
Hládek et al. [8], 2016	OCR	English	HMM, LM	Ristad and Yianilos [100]
Silfverberg et al. [121], 2016	OCR	Finnish	N/a	WFST, Eger et al. [122], Lindén [123]
Abandah et al. [124], 2015	Diacritization	Arabic	N/a	Recurrent ANN
Hasan and Heger [125], 2015	Search query	English	LM	DLD, SMT
Lai et al. [126], 2015	General, medical	English	CRF for NER	Kernighan et al. [22]
Ramasamy et al. [127], 2015	General	Czech	Golding and Roth [96]	Church and Gale [128]
Evershed and Fitch [129], 2014	OCR	English	LM	LCM
Makazhanov et al. [130], 2014	General	Kazakh	N/a	Church and Gale [128]

Note: ANN, artificial neural network; BERT, bidirectional encoder representations from transformers; BiLSTM, bidirectional long short-term memory; CRF, conditional random fields; HMM, hidden Markov model; LCM, letter-confusion matrix; LSTM, long short-term memory; NER, named entity recognition; OCR, optical character recognition; POS, part-of-speech; seq2seq, sequence-to-sequence; WFST, weighted finite-state transducer.

Table 6. spelling correction systems with learning of context model and error model—part II.

Reference	Application	Language	Context Model	Error Model
Mitankin et al. [131], 2014	OCR, historical	Old English	ME, LM	SMT
Sariev et al. [132], 2014	Historical text, OCR	Early Modern English, Bulgarian	SMT, LM, ME	LD, SMT
Wang et al. [133], 2014	Search query	English	N/a	ME
Huang et al. [134], 2013	General, automotive	English	N/a	Maximum of common characters, LD, ANN
Reffle and Ringlstetter [135], 2013	OCR, historical	Old German	LM	Bayes
Duan et al. [136], 2012	Search query	English	LM	SVM
Rashwan et al. [137], 2011	Diacritization	Arabic	LM	FSA, A star, ME
Perez-Cortes et al. [105], 2010	OCR, record linkage	Spanish	N/a	WFST, generalized LD, letter *n*-grams
Takasu [138], 2009	OCR	Japanese	N/a	Ristad and Yianilos [100], Takasu and Aihara [139]
Magdy and Darwish [140], 2008	OCR	Arabic	LM	LCM
Beaufort and Mancas-Thillou [141], 2007	OCR	English	WFST	LCM
Byun et al. [142], 2007	General	Korean	N/a	Learning general edit operations
Magdy and Darwish [143], 2006	OCR	Arabic	LM	Brill and Moore [29]
Oncina and Sebban [144], 2006	OCR	None	N/a	Ristad and Yianilos [100]
Ahmad and Kondrak [108], 2005	Search query	English	LM	Ristad and Yianilos [100]
Toutanova and Moore [6], 2002	General	English	N/a	Brill and Moore [29]
Brill and Moore [29], 2000	General	English	LM	DLD, extended Church and Gale [128]
Ristad and Yianilos [100], 1998	General	English	N/a	LCM
Church and Gale [128], 1991	General	English	N/a	Four LCMs
Kernighan et al. [22], 1990	General	English	N/a	Four LCMs

Note: ANN, artificial neural network; FSA, finite-state automaton; LCM, letter-confusion matrix; LM, language model; LSTM, long short-term memory; ME, maximum entropy; OCR, optical character recognition; seq2seq, sequence-to-sequence; SMT, statistical machine translation; SVM, support vector machine; WFST, weighted finite state transducer.

ASC systems with a learning error model often complement optical character recognition systems (OCR). The digitized document contains spelling errors characteristic of the quality of the paper, scanner, and OCR algorithm. If the training database (original and corrected documents) is large enough, the spelling system is adapted to the data. A training sample from the TREC-5 confusion track corpus [9] is displayed in Figure 7.

```
Correct:   bulletin
Incorrect: bM.etin ,bWetin bMetinh bUletin
Cunt:           2       2        4      23
```

Figure 7. Example misspellings of word the "bulletin" from optical character recognition (OCR).

7.1. Word-Confusion Set

The simplest method of estimating the learning error model is a word-confusion set that counts the cooccurrences of correct and incorrect words in the training corpus. It considers a pair of correct and incorrect words as one big edit operation. The word-confusion set remembers possible corrections for each frequently misspelled form (See Figure 7). This method was used by Gong et al. [145] to improve the precision of e-mail spam detection.

Its advantages are that it can be easily created and manually checked. The disadvantage of this simple approach is that it is not possible to obtain a corpus that has every possible misspelling for every possible word. The second problem of the word-confusion set is that error probabilities are far from "real" probabilities because training data are always sparse. Shannon's theorem states that it is not possible to be 100% accurate in spelling correction.

7.2. Learning String Metrics

The sparseness problems of the word-confusion set are solved by observing smaller subword units (such as letters or morphemes). For example, Makazhanov et al. [130] utilized information about morphemes in the Kazakh language to improve automatic spelling correction. The smallest possible subword units are letters. Estimating parameters of edit operations partially mitigates the sparseness problem because smaller sequences appear in the training corpus more frequently. The authors in [29] presented an error model that learned general edit operations. The antecedent and consequent parts of the edit operations can be arbitrary strings called partitions. The partition of the strings defines the edit operations.

Generalized edit distance is another form of a learning error model. The antecedent and consequent part of an edit operation is a single symbol that can be a letter or a special deletion mark. Edit distance is generalized by considering the arbitrary weight of an operation. Weights of each possible edit operation of the Levenshtein distance (LD) can be stored in a single letter-confusion matrix (LCM). $\boldsymbol{\Delta}$ weights for generalized edit distance are stored in four matrices [128]. The generalized edit distance is not always a metric in the strict mathematical sense because the distance in the opposite direction can be different. More theory about learning string metrics can be found in a book [146] or in a survey ([147], Section 5.1).

Weights $\boldsymbol{\Delta}$ in an LCM express the weight of error types (Figure 8). If the LCM is a matrix of ones with zeros on the main diagonal, it expresses the Levenshtein edit distance. Each edit operation has a value of 1, and the sum of edit operations is the Levenshtein edit distance. The edit distance with weights is calculated by a dynamic algorithm [53,148].

The LCM for a Levenshtein-like edit distance can be estimated with an expectation-maximization algorithm [100]. The learning algorithm calculates weights of operations for each training sample that are summed and normalized to form an updated letter confusion matrix.

If the training corpus is sparse (which it almost always is), the learning process brings the problem of overfitting. Hládek et al. [8] proposed a method for smoothing parameters in a letter-confusion matrix. Bilenko and Mooney [149] extended string-distance learning with an affine gap penalty

(allowing for random sequences of characters to be skipped). Also, Kim and Park [150] presented an algorithm for learning a letter-confusion matrix and for calculating generalized edit distance. This algorithm was further extended by Hyyrö et al. [151].

	a	b	c	d
a	1	0	0	0
b	0	1	0	0
c	0	0	1	0
d	0	0	0	1

	a	b	c~d	
a	0.2	0.1	0.2	0.3
b	0.2	0.5	0.2	0.3
c	0.2	0.1	0.6	0.3
d	0.2	0.1	0.2	0.8

Figure 8. Example of a letter-confusion matrix for the alphabet of symbols a, b, c, and d for Levenshtein distance (left) and arbitrary letter confusion matrix (right): the matrix gives a weight of transcription of the letter in the vertical axis to the letter in the horizontal axis.

7.3. Spelling Correction as Machine Translation of Letters

Spelling correction can be formulated as a problem of searching for the best transcription of an arbitrary sequence of symbols into another sequence. This type of problem can be solved with methods typical for machine translation. General string-to-string translation models are not restricted to the spelling error correction task but can also be applied to many problems, such as grapheme-to-phoneme conversion, transliteration, or lemmatization [122]. The machine-translation representation of the ASC overcomes the problem of joined and split words but requires a large corpus to properly learn the error model.

Zhou et al. [117] defined the machine-translation approach to spelling correction by the following equation:

$$s' = \arg\max_{s} P(s|S), \tag{4}$$

where S is the given incorrect sequence, s is the possibly correct sequence, and s' is the best correction.

Characters are “words" of “correct" and “incorrect" language. Words in the training database are converted into sequences of lowercase characters, and white spaces are converted into special characters. The machine-translation system is trained on a parallel corpus of examples of spelling errors and corrections. Sariev et al. [132] and Koehn et al. [152] proposed an ASC system that utilizes a statistical machine-translation system called Moses (http://www.statmt.org/moses/).

The authors in [125] cast spelling correction into the machine translation of character bigrams. The spelling system is trained on logs of search queries. It was assumed that the corrections of queries by the user follow misspelled queries. This heuristics creates a training database. To improve precision, character bigrams are used instead of single characters.

Statistical machine-translation models based on string alignment, translation phrases, and n-gram language models are replaced by neural machine-translation systems. The basic neural-translation architecture, based on a neural encoder and decoder, was proposed by Sutskever et al. [110]. The translation model learns $P(y_1..y_T|x_1...x_T)$ by encoding the given sequence into a fixed-size vector [117]:

$$s = f_e(x_1, ..., x_T) = h_T\,. \tag{5}$$

The sequence-embedding vector is decoded into another sequence by a neural decoder [117]:

$$y_t = f_d(s, y_1, ..., y_{t-1}) = h_T\,. \tag{6}$$

The decoder takes the encoded vector language model and generates the output. Zhou et al. [117] showed that, by using k-best decoding in the string-to-string translation models, they achieved much better results on the spelling correction task than those of the three baselines, namely edit distance, weighted edit distance, and the Brill and Moore model [104].

8. Evaluation Methods

The development of automatic spelling correction systems requires a way to objectively assess the results. It is clear though that it is impossible to propose a “general" spelling benchmark because the problem is language- and application-dependent.

Three possible groups of methods exist for evaluating automatic spelling correction:

- accuracy, precision, and recall (classification);
- bilingual-evaluation-understudy (BLEU) score (machine translation); and
- mean reciprocal rank and mean average precision (information retrieval).

The most common evaluation metrics is classification accuracy. The disadvantage of this method is that only the best candidate from the suggestion list is considered, and order and count of the other proposed correction candidates are insignificant. Therefore, it is not suitable for evaluating an interactive system.

Automatic spelling correction is similar to machine translation. A source text containing errors is translated to its most probable correct form. The approach takes the whole resulting sentence, and it is also convenient for evaluating the correction of a poor writing style and non-word errors. It was used by Sariev et al. [132], Gerdjikov et al. [153] and Mitankin et al. [131].

Machine-translation systems are evaluated using the BLEU score, which was first proposed by Papineni et al. [154]:

> “The task of a BLEU implementation is to compare *n*-grams of the candidate with the *n*-grams of the reference translation and to count the number of matches. These matches are position-independent. The more matches, the better the candidate translation.”

The process of automatic spelling correction is also similar to information retrieval. An incorrect word is a query, and the sorted list of the correction candidates is the response. This approach evaluates the whole list of suggestions and favors small lists of good (highly ranked) candidates for correction. The two following evaluation methodologies are used to evaluate spelling:

- Mean reciprocal rank: A statistical measure for evaluating any process that produces a list of possible responses to a sample of queries ordered by the probability of correctness. The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries [155].
- Mean average precision: Average precision observes how many times a correct suggestion is on the *n*-th place or better in a candidate list [40]. It is calculated from average precision for *n* in the range from 1 to *k* (*k* is a constant, e.g., 10).

Machine translation and information retrieval are well-suited for evaluating interactive systems because they consider the whole candidate list. A smaller candidate list is more natural to comprehend. The best correction can be selected faster from fewer words. On the other hand, the candidate list must be large enough to contain the correct answer.

8.1. Evaluation Corpora and Benchmarks

Several authors proposed corpora for specific tasks and languages, but no approach was broadly accepted. The authors in [12] proposed the Koran as a benchmark for the evaluation of Arabic diacritizations. Reynaert [156] presented an XML format and OCR-processed historical document set in Dutch for the evaluation of automatic spelling correction systems.

The most used evaluation set for automatic spelling correction of OCR is TREC-5 Confusion Track [9]. It was created by scanning a set of paper documents. The database consists of original and recognized documents, so it is possible to identify correct–incorrect pairs for system training and

evaluation. The other common evaluation set is Microsoft Speller Challenge (https://www.microsoft.com/en-us/download/details.aspx?id=52351).

Also, Hagen et al. [34] proposed a corpus of corrected search queries in English (https://www.uni-weimar.de/en/media/chairs/computer-science-and-media/webis/corpora), and provided an evaluation metric. They re-implemented the best-performing approach [157] from the Microsoft Speller Challenge (https://github.com/webis-de/SIGIR-17).

Tseng et al. [158] presented a complete publicly available spelling benchmark for the Chinese language, preceded by Wu et al. [159]. Similarly, the first competition on automatic spelling correction for Russian was published by Sorokin et al. [160].

8.2. Performance Comparison

Table 7 gives a general overview of the performance of automatic spelling correction systems. It lists approaches with well-defined evaluation experiments performed by the authors. The table displays the best value reached in the evaluation and summarizes the evaluation corpora. Only a few corpora were available that are suitable for evaluating an ASC system (such as TREC-5).

It is virtually impossible to compare the performance of state-of-the-art spelling correction systems. Each author solves a different task and uses their methodology, custom testing set, and various evaluation corpora with different languages. The displayed values cannot be used for mutual comparison but are instead a guide for selecting an evaluation method. A solution would be a spelling correction toolkit that implements state-of-the-art methods for error modeling and context classification. A standard set of tools would allow for comparison of individual components, such as error models.

Table 7. Reported evaluation results.

Approach	Evaluation (%)	Test Corpus
Azmi et al. [60], 2019	A 98, F 90.7, P 83.5, R 99.2	Arabic Newspaper Corpora
Han et al. [109], 2019	A 62.67, F 49.33, P 81.12, R 36.33	Tseng et al. [158], Wu et al. [159]
Lv et al. [72], 2016	A 95.72, F 95.78	Chinese OCR Medical Records
Melero et al. [73], 2016	P 82.56	Twitter texts (baseline 56.88%)
Attia et al. [107], 2015	A 93.64	Arabic Gigaword Corpus 5th Edition
Lai et al. [126], 2015	A 88.2, F 94.4, P 96.2, R 92.7	Clinical Notes of Patients
Ramasamy et al. [127], 2015	F 95.4, P 95.0, R 95.9	WebColl, CzeSL-MAN, Czech National Corpus: SYN2005 and SYN2010
Evershed and Fitch [129], 2014	W 6.4	Sydney Morning Herald, 1842-1954
Mitankin et al. [131], 2014	A 93.96	1641 Depositions Old English
Sariev et al. [132], 2014	W 16.84/4.98/4.25/3.27	ICAMET/IMPACT BG/1641 Deposition/TREC-5
Sagiadinos et al. [79], 2014	F 97.4, P 97.8, R 97.0	Eleftherotypia—The Modern Greek Text Corpus
Wang et al. [133], 2014	F 85.89	Microsoft Speller Challenge
Ehsan and Faili [80], 2013	F 36, P 56, R 31	Persian Corpus Peykareh
Duan et al. [136], 2012	F 94.9/92.8, P 96.3/90.3, R 94.4/95.3	TREC-5/Microsoft Speller Challenge
Flor [59], 2012	F 85.87, P 85.50, R 86.25	ETS Spelling Corpus
Sha et al. [42], 2011	A 93.3	User Behavior Records in Real Online Study Website Chinese
Stüker et al. [85], 2011	W 9.7	The Fay Database—Children's Free Writing German
Wong and Glance [86], 2011	A 88.73	Clinical Progress Notes (http://physionet.org)
Takasu [138], 2009	A 94.2	1000 Japanese Articles
Beaufort and Mancas-Thillou [141], 2007	A 65.4	English ICDAR 2003 Corpus
Byun et al. [142], 2007	A 92.75	SMS messages in Korean
Carlson and Fette [89], 2007	A 95.2/95.8	Brown Corpus/Wall Street Journal Corpus
Magdy and Darwish [143], 2006	W 11.7	Arabic Book "The Provisions of the Return" (2000 words)
van Delden et al. [46], 2004	A 93.3	Misspellings from two NASA databases, Structural and Fuel Cells
Héja and Surján [92], 2003	P 37.2, R 82.6	Corpus of 92 Clinical Diagnoses in Hungarian

Note: A, accuracy; F, F-measure/F1-score; P, precision; R, recall; W, word error rate.

9. Conclusions

The chronological sorting and grouping of the summary tables with references in this work reveal several findings. The research since the last comprehensive survey [2] brought new methods for spelling correction. On the other hand, we can say that the progress of spelling correction in all areas was slow until the introduction of deep neural networks.

New, a priori spelling correction systems are often presented for low-resource languages. Authors propose rules for a priori error model that extend the existing phonetic algorithm or adjust the edit distance for the specifics of the given language.

Spelling correction systems in context are mostly proposed for languages with sufficient language resources for language modeling. Most of them use n-gram language models, but some approaches use neural networks or other classifiers. Scientific contributions for spelling in context explore various context features with statistical classifiers.

Spelling correction with the learning error model shows the biggest progress. The attention of the researchers moves from statistical estimation of the letter confusion matrices to utilization of the statistical machine translation.

This trend is visible mainly in Tables 5 and 6, where we can observe the growing popularity of the use of encoder–decoder architecture and deep neural networks since 2018. New approaches move from word-level correction to arbitrary character sequence correction because new methods based on deep neural networks bring better possibilities. Methods based on machine translation and deep learning solve the weakest points of the ASC systems, such as language-specific rules, real-word errors, and spelling errors with spaces. The neural networks can be trained on already available large textual corpora.

The definition of the spelling correction stated in the Equation (1) begins to be outdated because of the new methods. Classical statistical models of context-based (n-gram, log-linear regression, and naïve Bayes classifier) on the presence of word-level features in the context are no longer important. Instead, feature extraction is left to the hidden layers of the deep neural network. The correction of spelling errors becomes the task of transcribing a sequence of characters to another sequence of characters using a neural network, as it is stated in Equation (4). Research in the field of spelling error correction thus approaches other solutions to other tasks of speech and language processing, such as machine translation or fluent speech recognition.

On the other hand, the scientific progress of learning error models is restricted by the lack of training corpora and evaluation benchmarks. Our examination of the literature shows that there is no consensus on how to evaluate and compare spelling correction systems. Instead, almost every paper uses its own evaluation set and evaluation methodology. In our opinion, the reason is that most of the spelling approaches strongly depend on the specifics of the language and are hard to adapt to another language or a different application. Recent algorithms based on deep neural networks are not language dependent, but their weak point is that they require a large training set, often with expensive manual annotation. These open issues call for new research in automatic spelling correction.

Author Contributions: Conceptualization, D.H.; methodology, D.H.; formal analysis, J.S.; investigation, D.H.; resources, D.H.; writing—original draft preparation, D.H.; writing—review and editing, M.P. and J.S.; supervision, M.P.; project administration, J.S.; funding acquisition, M.P. All authors have read and agreed to the published version of the manuscript.

Funding: Research in this paper was supported by the Slovak Research and Development Agency (Agentúra na podporu výskumu a vývoja) under projects APVV-15-0517 and APVV-15-0731; the Scientific Grant Agency (Vedecká grantová agentúra MŠVVaŠ SR a SAV), project number VEGA 1/0753/20; and the Cultural and Educational Grant Agency (Kultúrna a edukačná grantová agentúra MŠVVaŠ SR), project number KEGA 009TUKE-4-2019, both funded by the Ministry of Education, Science, Research, and Sport of the Slovak Republic.

Acknowledgments: The authors want to thank Jozef Juhár for the team leadership, and personal and financial support.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations were used in this manuscript:

ANN	Artificial neural network
ASC	Automatic spelling correction
CFG	Context-free grammar
CRF	Conditional random fields
DLD	Damerau–Levenshtein distance
EM	Expectation maximization
FSA	Finite-state automaton
HMM	Hidden Markov model
IR	Information retrieval
k-NN	k-nearest neighbors
LCS	Longest common subsequence
LD	Levenshtein distance
LCM	Learning letter-confusion matrix
LM	Language model
LR	Linear regression
LSA	Latent semantic analysis
LSTM	Long short-term memory
seq2seq	Sequence-to-sequence
ME	Maximum entropy
OCR	Optical character recognition
PMI	Pointwise mutual information
POS	Part-of-speech tagging
RF	Random forests
SMT	Statistical machine translation
SVM	Support vector machine
WFST	Weighted finite-state transducer
WSD	Word-sense disambiguation
WCS	Word-confusion set

References

1. Cai, F.; de Rijke, M. A Survey of Query Auto Completion in Information Retrieval. *Found. Trends Inf. Retr.* **2016**, *10*, 273–363. [CrossRef]
2. Kukich, K. Techniques for automatically correcting words in text. *Acm Comput. Surv.* **1992**, *24*, 377–439.
3. Baba, Y.; Suzuki, H. How are spelling errors generated and corrected? A study of corrected and uncorrected spelling errors using keystroke logs. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju Island, Korea, 8–14 July 2020; pp. 373–377.
4. Mitton, R. *English Spelling and the Computer*; Longman Group: Harlow, Essex, UK, 1996; p. 214.
5. Yannakoudakis, E.J.; Fawthrop, D. The rules of spelling errors. *Inf. Process. Manag.* **1983**, *19*, 87–99. [CrossRef]
6. Toutanova, K.; Moore, R.C. Pronunciation Modeling for Improved Spelling Correction. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, USA, 7–12 July 2002; pp. 144–151. [CrossRef]
7. Pirinen, T.A.; Lindén, K. State-of-the-art in weighted finite-state spell-checking. In *Computational Linguistics and Intelligent Text Processing, Proceedings of the CICLing 2014, Kathmandu, Nepal, 6–12 April 2014*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8404, Part 2, pp. 519–532. [CrossRef]
8. Hládek, D.; Staš, J.; Ondáš, S.; Juhár, J.; Kovács, L. Learning string distance with smoothing for OCR spelling correction. *Multimed. Tools Appl.* **2017**, *76*, 24549–24567. [CrossRef]
9. Kantor, P.B.; Voorhees, E.M. The TREC-5 Confusion Track: Comparing Retrieval Methods for Scanned Text. *Inf. Retr.* **2000**, *2*, 165–176. [CrossRef]

10. Gimenes, P.A.; Roman, N.T. Spelling error patterns in Brazilian Portuguese. *Comput. Linguist.* **2015**, *41*, 175–184. [CrossRef]
11. Zitouni, I.; Sarikaya, R. Arabic diacritic restoration approach based on maximum entropy models. *Comput. Speech Lang.* **2009**, *23*, 257–276. [CrossRef]
12. Azmi, A.M.; Almajed, R.S. A survey of automatic Arabic diacritization techniques. *Nat. Lang. Eng.* **2015**, *21*, 477–495. [CrossRef]
13. Asahiah, F.O.; Odéjobi, O.A.; Adagunodo, E.R. A survey of diacritic restoration in abjad and alphabet writing systems. *Nat. Lang. Eng.* **2018**, *24*, 123–154. [CrossRef]
14. Miangah, T.M. FarsiSpell: A spell-checking system for Persian using a large monolingual corpus. *Lit. Linguist. Comput.* **2014**, *29*, 56–73. [CrossRef]
15. Shang, H.; Merrettal, T. Tries for approximate string matching. *IEEE Trans. Knowl. Data Eng.* **1996**, *8*, 540–547. [CrossRef]
16. Pal, U.; Kundu, P.K.; Chaudhuri, B.B. OCR error correction of an inflectional Indian language using morphological parsing. *J. Inf. Sci. Eng.* **2000**, *16*, 903–922.
17. Mashod Rana, M.; Tipu Sultan, M.; Mridha, M.F.; Eyaseen Arafat Khan, M.; Masud Ahmed, M.; Abdul Hamid, M. Detection and Correction of Real-Word Errors in Bangla Language. In Proceedings of the 2018 International Conference on Bangla Speech and Language Processing, ICBSLP 2018, Sylhet, Bangladesh, 21–22 September 2018; pp. 1–4. [CrossRef]
18. Boytsov, L. Indexing methods for approximate dictionary searching. *J. Exp. Algorithmics* **2011**, *16*, 11–91. [CrossRef]
19. Deorowicz, S.; Ciura, M.G. Correcting spelling errors by modelling their causes. *Int. J. Appl. Math. Comput. Sci.* **2005**, *15*, 275–285.
20. Wang, Y.R.; Liao, Y.F. Word vector/conditional random field-based Chinese spelling error detection for SIGHAN-2015 evaluation. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, Beijing, China, 30–31 July 2015; pp. 46–49. [CrossRef]
21. Zhang, H.; Zhang, Q. EmbedJoin: Efficient edit similarity joins via embeddings. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and lData Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 585–594. [CrossRef]
22. Kernighan, M.D.; Church, K.W.; Gale, W.A. A spelling correction program based on a noisy channel model. In Proceedings of the 13th Conference on Computational Linguistics, Helsinki, Finland, 20–25 August 1990; pp. 205–210. [CrossRef]
23. Jurafsky, D.; Martin, J.H. *Speech and Language Processing*; Prentice Hall: Upper Saddle River, NJ, USA, 2014.
24. Kinaci, A.C. Spelling Correction Using Recurrent Neural Networks and Character Level N-gram. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018, Malatya, Turkey, 28–30 September 2018. [CrossRef]
25. Reffle, U. Efficiently generating correction suggestions for garbled tokens of historical language. *Nat. Lang. Eng.* **2011**, *17*, 265–282. [CrossRef]
26. Yu, M.; Li, G.; Deng, D.; Feng, J. String similarity search and join: A survey. *Front. Comput. Sci.* **2016**, *10*, 399–417. [CrossRef]
27. Vilares, M.; Otero, J.; Barcala, F.M.; Domínguez, E.; Dominguez, E. Automatic spelling correction in Galician. In *Advances in Natural Language Processing*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3230, pp. 45–57.
28. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 623–656. [CrossRef]
29. Brill, E.; Moore, R.C. An improved error model for noisy channel spelling correction. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics ACL 00, Hong Kong, China, 7 October 2000; pp. 286–293. [CrossRef]
30. Khairul Islam, M.I.; Meem, R.I.; Abul Kasem, F.B.; Rakshit, A.; Habib, M.T. Bangla Spell Checking and Correction Using Edit Distance. In Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019, Dhaka, Bangladesh, 3–5 May 2019. [CrossRef]
31. Hawezi, R.S.; Azeez, M.Y.; Qadir, A.A. Spell checking algorithm for agglutinative languages 'Central Kurdish as an example'. In Proceedings of the 5th International Engineering Conference, IEC 2019, Erbil, Iraq, 23–25 June 2019; pp. 142–146. [CrossRef]

32. Thaiprayoon, S.; Kongthon, A.; Haruechaiyasak, C. ThaiQCor 2.0: Thai Query Correction via Soundex and Word Approximation. In Proceedings of the ICAICTA 2018—5th International Conference on Advanced Informatics: Concepts Theory and Applications, Krabi, Thailand, 14–17 August 2018; pp. 113–117. [CrossRef]
33. Christanti, M.V.; Naga, D.S. Fast and accurate spelling correction using trie and Damerau-levenshtein distance bigram. *Telkomnika (Telecommun. Comput. Electron. Control.)* **2018**, *16*, 827–833. [CrossRef]
34. Hagen, M.; Potthast, M.; Gohsen, M.; Rathgeber, A.; Stein, B. A large-scale query spelling correction corpus. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017. [CrossRef]
35. Sakuntharaj, R.; Mahesan, S. A novel hybrid approach to detect and correct spelling in Tamil text. In Proceedings of the 2016 IEEE International Conference on Information and Automation for Sustainability: Interoperable Sustainable Smart Systems for Next Generation, ICIAfS 2016, Galle, Sri Lanka, 16–19 December 2016. [CrossRef]
36. Vobl, T.; Gotscharek, A.; Reffle, U.; Ringlstetter, C.; Schulz, K.U. PoCoTo—an open source system for efficient interactive postcorrection of OCRed historical texts. In Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage—DATeCH '14, Brussels, Belgium, 8–10 May 2019. [CrossRef]
37. Rees, T. Taxamatch, an algorithm for near ('Fuzzy') matching of scientific names in taxonomic databases. *PLoS ONE* **2014**, *9*, e107510. [CrossRef]
38. Mühlberger, G.; Zelger, J.; Sagmeister, D. User-driven correction of OCR errors: combing crowdsourcing and information retrieval technology. In Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage—DATeCH '14, Brussels, Belgium, 8–10 May 2019. [CrossRef]
39. Patrick, J.; Nguyen, D. Automated Proof Reading of Clinical Notes. In Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation (PACLIC 25), Singapore, 16–18 December 2011; pp. 303–312.
40. Kashefi, O.; Sharifi, M.; Minaie, B. A novel string distance metric for ranking Persian respelling suggestions. *Nat. Lang. Eng.* **2013**, *19*, 259–284.
41. Andrade, G.; Teixeira, F.; Xavier, C.R.; Oliveira, R.S.; Rocha, L.C.; Evsukoff, A.G. HASCH: High performance automatic spell checker for portuguese texts from the web. *Procedia Comput. Sci.* **2012**, *9*, 403–411. [CrossRef]
42. Sha, S.; Jun, L.; Qinghua, Z.; Wei, Z. Automatic Chinese Topic Term Spelling Correction in Online Pinyin Input. In Proceedings of the International Conference on Human-centric Computing 2011 and Embedded and Multimedia Computing 2011, Enshi, China, 11–13 August 2011; pp. 23–36. [CrossRef]
43. Naji, N.; Savoy, J. Information retrieval strategies for digitized handwritten medieval documents. In *Asia Information Retrieval Symposium—AIRS 2011: Information Retrieval Technology*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7097, pp. 103–114. [CrossRef]
44. Bustamante, F.R.; Arnaiz, A.; Ginés, M. A spell checker for a world language: The new Microsoft's Spanish spell checker. In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy, 22–28 May 2006; pp. 83–86.
45. UzZaman, N.; Khan, M. A Double Metaphone encoding for Bangla and its application in spelling checker. In Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, IEEE NLP-KE'05, Wuhan, China, 30 October–1 November 2005; Volume 2005, pp. 705–710. [CrossRef]
46. van Delden, S.; Bracewell, D.; Gomez, F. Supervised and unsupervised automatic spelling correction algorithms. In Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, IRI 2004, Las Vegas, NV, USA, 8–10 November 2004; pp. 530–535. [CrossRef]
47. Schulz, K.U.; Mihov, S. Fast string correction with Levenshtein automata. *Int. J. Doc. Anal. Recognit.* **2003**, *5*, 67–85. [CrossRef]
48. Taghva, K.; Stofsky, E. OCRSpell: An interactive spelling correction system for OCR errors in text. *Int. J. Doc. Anal. Recognit.* **2001**, *3*, 125–137. [CrossRef]
49. Vagelatos, A.; Triantopoulou, T.; Tsalidis, C.; Christodoulakis, D. Utilization of a lexicon for spelling correction in modern Greek. In Proceedings of the 1995 ACM symposium on Applied computing—SAC '95, Nashville, TN, USA, 26–28 February 1995; pp. 267–271. [CrossRef]

50. Ahmed, F.; de Luca, E.W.; Nürnberger, A. Revised N-Gram based Automatic Spelling Correction Tool to Improve Retrieval Effectiveness. *Polibits* **2009**, *40*, 39–48. [CrossRef]
51. Levenshtein, V.I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.
52. Damerau, F.J. A Technique for Computer Detection and Correction of Spelling Errors. *Commun. ACM* **1964**, *7*, 171–176. [CrossRef]
53. Wagner, R.a.; Fischer, M.J. The String-to-String Correction Problem. *J. ACM* **1974**, *21*, 168–173. [CrossRef]
54. Flouri, T.; Giaquinta, E.; Kobert, K.; Ukkonen, E. Longest common substrings with k mismatches. *Inf. Process. Lett.* **2015**, *115*, 643–647. [CrossRef]
55. Bergroth, L.; Hakonen, H.; Raita, T. A survey of longest common subsequence algorithms. In Proceedings of the 7th International Symposium on String Processing and Information Retrieval, SPIRE 2000, A Coruña, Spain, 27–29 September 2000; pp. 39–48. [CrossRef]
56. Naseem, T.; Hussain, S. A novel approach for ranking spelling error corrections for Urdu. *Lang. Resour. Eval.* **2007**, *41*, 117–128. [CrossRef]
57. Philips, L. Hanging on the metaphone. *Comput. Lang.* **1990**, *7*, 38–44.
58. Kondrak, G.; Sherif, T. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In Proceedings of the Workshop on Linguistic Distances—LD '06, Sydney, Australia, 23 July 2006; pp. 43–50. [CrossRef]
59. Flor, M. Four types of context for automatic spelling correction. *TAL Trait. Autom. Des Langues* **2012**, *53*, 61–99.
60. Azmi, A.M.; Almutery, M.N.; Aboalsamh, H.A. Real-Word Errors in Arabic Texts: A Better Algorithm for Detection and Correction. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 1308–1320. [CrossRef]
61. Dong, R.; Yang, Y.; Jiang, T. Spelling correction of non-word errors in Uyghur-Chinese machine translation. *Information* **2019**, *10*, 202. [CrossRef]
62. Yazdani, A.; Ghazisaeedi, M.; Ahmadinejad, N.; Giti, M.; Amjadi, H.; Nahvijou, A. Automated Misspelling Detection and Correction in Persian Clinical Text. *J. Digit. Imaging* **2019**, *33*, 555–562. [CrossRef]
63. Damnati, G.; Auguste, J.; Nasr, A.; Charlet, D.; Heinecke, J.; Béchet, F. Handling normalization issues for part-of-speech tagging of online conversational text. In Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; pp. 88–92.
64. Dashti, S.M.S. Real-word error correction with trigrams: correcting multiple errors in a sentence. *Lang. Resour. Eval.* **2018**, *52*, 485–502. [CrossRef]
65. Fahda, A.; Purwarianti, A. A statistical and rule-based spelling and grammar checker for Indonesian text. In Proceedings of the 2017 International Conference on Data and Software Engineering, ICoDSE 2017, Palembang, Indonesia, 1–2 November 2017; pp. 1–6. [CrossRef]
66. Heyman, G.; Vulić, I.; Laevaert, Y.; Moens, M.F. Automatic detection and correction of context-dependent dt-mistakes using neural networks. *Comput. Linguist. Neth. J.* **2018**, *8*, 49–65.
67. Golding, A.R.; Schabes, Y. Combining Trigram-based and feature-based methods for context-sensitive spelling correction. In Proceedings of the 34th annual meeting on Association for Computational Linguistics, Santa Cruz, CA, USA, 23–28 June 1996; pp. 71–78. [CrossRef]
68. Dziadek, J.; Henriksson, A.; Duneld, M. Improving Terminology Mapping in Clinical Text with Context-Sensitive Spelling Correction. *Stud. Health Technol. Inform.* **2017**, *235*, 241–245. [CrossRef]
69. Sorokin, A. Spelling Correction for Morphologically Rich Language: A Case Study of Russian. In Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing, Valencia, Spain, 4 April 2017; pp. 45–53. [CrossRef]
70. Zhao, H.; Cai, D.; Xin, Y.; Wang, Y.; Jia, Z. A Hybrid Model for Chinese Spelling Check. *ACM Trans. Asian -Low-Resour. Lang. Inf. Process.* **2017**, *16*, 1–22. [CrossRef]
71. de Mendonça Almeida, G.A.; Avanço, L.; Duran, M.S.; Fonseca, E.R.; Volpe Nunes, M.d.G.; Aluísio, S.M. Evaluating phonetic spellers for user-generated content in Brazilian Portuguese. In Proceedings of the PROPOR 2016: Computational Processing of the Portuguese Language, Tomar, Portugal, 13–15 July 2016; pp. 361–373. [CrossRef]
72. Lv, Y.Y.; Deng, Y.I.; Liu, M.L.; Lu, Q.Y. Automatic error checking and correction of electronic medical records. *Front. Artif. Intell. Appl.* **2016**, *281*, 32–40. [CrossRef]

73. Melero, M.; Costa-Jussà, M.; Lambert, P.; Quixal, M. Selection of correction candidates for the normalization of Spanish user-generated content. *Nat. Lang. Eng.* **2016**, *22*, 135–161. [CrossRef]
74. Mirzababaei, B.; Faili, H. Discriminative reranking for context-sensitive spell-checker. *Digit. Scholarsh. Humanit.* **2016**, *31*, 411–427. [CrossRef]
75. Sorokin, A.; Shavrina, T. Automatic spelling correction for Russian social media texts. In Proceedings of the International Conference "Dialogue 2016", Moscow, Russia, 1–4 June 2016; pp. 688–701.
76. Vilares, J.; Alonso, M.A.; Doval, Y.; Vilares, M. Studying the effect and treatment of misspelled queries in Cross-Language Information Retrieval. *Inf. Process. Manag.* **2016**, *52*, 646–657. [CrossRef]
77. Lhoussain, A.S.; Hicham, G.; Abdellah, Y. Adaptating the levenshtein distance to contextual spelling correction. *Int. J. Comput. Sci. Appl.* **2015**, *12*, 127–133.
78. Ferrero, C.L.; Renau, I.; Nazar, R.; Torner, S. Computer-assisted Revision in Spanish Academic Texts: Peer-assessment. *Procedia-Soc. Behav. Sci.* **2014**, *141*, 470–483. [CrossRef]
79. Sagiadinos, S.; Gasteratos, P.; Dragonas, V.; Kalamara, A.; Spyridonidou, A.; Kermanidis, K. Knowledge-Poor Context-Sensitive Spelling Correction for Modern Greek. In *Artificial Intelligence: Methods and Applications*; Springer International Publishing: Cham, Switzerland, 2014; Volume 8445, pp. 360–369. [CrossRef]
80. Ehsan, N.; Faili, H. Grammatical and context-sensitive error correction using a statistical machine translation framework: Grammar and Context-Sensitive Error Checker. *Softw. Pract. Exp.* **2013**, *43*, 187–206. [CrossRef]
81. Hladek, D.; Stas, J.; Juhar, J.; Hládek, D.; Staš, J.; Juhar, J.; Hladek, D. Unsupervised spelling correction for Slovak. *Adv. Electr. Electron. Eng.* **2013**, *11*, 392–397. [CrossRef]
82. Bouma, G. Normalized (Pointwise) Mutual Information in Collocation Extraction. In Proceedings of the German Society for Computational Linguistics (GSCL 2009), Darmstadt, Germany, 25–27 September 2009; pp. 31–40.
83. Alkanhal, M.I.; Al-Badrashiny, M.A.; Alghamdi, M.M.; Al-Qabbany, A.O. Automatic stochastic arabic spelling correction with emphasis on space insertions and deletions. *IEEE Trans. Audio, Speech Lang. Process.* **2012**, *20*, 2111–2122. [CrossRef]
84. Grozea, C. Experiments and Results with Diacritics Restoration in Romanian. In Proceedings of the 15th International Conference on Text, Speech and Dialogue, TSD 2012, Brno, Czech Republic, 3–7 September 2012; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7499 LNAI, pp. 199–206. [CrossRef]
85. Stüker, S.; Fay, J.; Berkling, K. Towards Context-Dependent Phonetic Spelling Error Correction in Children's Freely Composed Text for Diagnostic and Pedagogical Purposes. In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2011, Florence, Italy, 27–31 August 2011; pp. 1601–1604.
86. Wong, W.; Glance, D. Statistical semantic and clinician confidence analysis for correcting abbreviations and spelling errors in clinical progress notes. *Artif. Intell. Med.* **2011**, *53*, 171–180. [CrossRef]
87. Abdulkader, A.; Casey, M.R. Low cost correction of OCR errors using learning in a multi-engine environment. In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, Barcelona, Spain, 26–29 July 2009; pp. 576–580. [CrossRef]
88. Farooq, F.; Bhardwaj, A.; Govindaraju, V. Using topic models for OCR correction. *Int. J. Doc. Anal. Recognit.* **2009**, *12*, 153–164. [CrossRef]
89. Carlson, A.; Fette, I. Memory-based context-sensitive spelling correction at web scale. In Proceedings of the 6th International Conference on Machine Learning and Applications, ICMLA 2007, Cincinnati, OH, USA, 13–15 December 2007; pp. 166–171. [CrossRef]
90. Banko, M.; Brill, E. Scaling to Very Very Large Corpora for Natural Language Disambiguation. In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, Toulouse, France, 5–10 July 2011; Association for Computational Linguistics: Stroudsburg, PA, USA, 2001; pp. 26–33. [CrossRef]
91. Mykowiecka, A.; Marciniak, M. Domain-driven automatic spelling correction for mammography reports. *Adv. Soft Comput.* **2006**, *35*, 521–530.
92. Héja, G.; Surján, G. Using N-gram method in the decomposition of compound medical diagnoses. *Stud. Health Technol. Inform.* **2002**, *90*, 455–459. [CrossRef]

93. Jin, R.; Zhai, C.; Hauptmann, A.G. Information retrieval for OCR documents: A content-based probabilistic correction model. *Proc. SPIE— Int. Soc. Opt. Eng.* **2003**, *5010*, 128–135. [CrossRef]
94. Ruch, P.; Baud, R.; Geissbühler, A. Using lexical disambiguation and named-entity recognition to improve spelling correction in the electronic patient record. *Artif. Intell. Med.* **2003**, *29*, 169–184. [CrossRef]
95. Li, J.; Wang, X. Combining trigram and automatic weight distribution in Chinese spelling error correction. *J. Comput. Sci. Technol.* **2002**, *17*, 915–923. [CrossRef]
96. Golding, A.R.; Roth, D. A Winnow-Based Approach to Context-Sensitive Spelling Correction. *Mach. Learn.* **1999**, *34*, 107–130. [CrossRef]
97. Carlson, A.J.; Rosen, J.; Roth, D. Scaling Up Context-Sensitive Text Correction. In Proceedings of the Thirteenth Conference on Innovative Applications of Artificial Intelligence Conference, Seattle, WA, USA, 7–9 August 2001; AAAI Press: Palo Alto, CA, USA, 2001; Volume 51, pp. 45–50. [CrossRef]
98. Ruch, P.; Baud, R.; Geissbuhler, A. Toward filling the gap between interactive and fully-automatic spelling correction using the linguistic context. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Tucson, AZ, USA, 7–10 October 2001; Volume 1, pp. 199–204. [CrossRef]
99. Jones, M.P.; Martin, J.H. Contextual spelling correction using latent semantic analysis. In Proceedings of the Fifth Conference on Applied Natural Language Processing—ANLC '97, Washington, DC, USA, 31 March–3 April 1997; Association for Computational Linguistics: Stroudsburg, PA, USA, 1997; pp. 166–173. [CrossRef]
100. Ristad, E.S.; Yianilos, P.N. Learning string-edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 522–532. [CrossRef]
101. Fivez, P.; Šuster, S.; Daelemans, W. Unsupervised context-sensitive spelling correction of English and Dutch clinical free-text with word and character N-Gram embeddings. *Comput. Linguist. Neth. J.* **2017**, *7*, 39–52.
102. Zaky, D.; Romadhony, A. An LSTM-based Spell Checker for Indonesian Text. In Proceedings of the 2019 International Conference on Advanced Informatics: Concepts, Theory, and Applications, ICAICTA 2019, Yogyakarta, Indonesia, 20–21 September 2019. [CrossRef]
103. Littlestone, N. Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. *Mach. Learn.* **1988**, *2*, 285–318. [CrossRef]
104. Brill, E.; Brill, E.; Wu, J.; Wu, J. Classifier Combination for Improved Lexical Disambiguation. In Proceedings of the 17th International Conference on Computational Linguistics, Montreal, QC, Canada, 10–14 August 1998; Association for Computational Linguistics: Stroudsburg, PA, USA, 1998; Volume 1, pp. 191–195. [CrossRef]
105. Perez-Cortes, J.C.; Llobet, R.; Navarro-Cerdan, J.R.; Arlandis, J. Using field interdependence to improve correction performance in a transducer-based OCR post-processing system. In Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition, ICFHR 2010, Kolkata, India, 16–18 November 2010; pp. 605–610. [CrossRef]
106. Attia, M.; Pecina, P.; Toral, A.; Tounsi, L.; van Genabith, J. An open-source finite state morphological transducer for modern standard Arabic. In Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing, Blois, France, 12–15 July 2011; Association for Computational Linguistics: Stroudsburg, PA, USA, 2011; pp. 125–133.
107. Attia, M.; Pecina, P.; Samih, Y.; Shaalan, K.; Van Genabith, J.; Genabith, J.V. Arabic spelling error detection and correction. *Nat. Lang. Eng.* **2015**, *22*, 1–23. [CrossRef]
108. Ahmad, F.; Kondrak, G. Learning a spelling error model from search query logs. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing HLT 05, Vancouver, BC, Canada, 6–8 October2005; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005; pp. 955–962. [CrossRef]
109. Han, Z.; Lv, C.; Wang, Q.; Fu, G. Chinese Spelling Check based on Sequence Labeling. In Proceedings of the 2019 International Conference on Asian Language Processing, IALP 2019, Shanghai, China, 15–17 November 2019; pp. 373–378. [CrossRef]
110. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*; MIT Press: Cambridge, MA, USA, 2014; pp. 3104–3112. [CrossRef]

111. Jain, A.; Jain, M.; Jain, G.; Tayal, D.K. "UTTAM": An efficient spelling correction system for Hindi language based on supervised learning. *ACM Trans. Asian -Low-Resour. Lang. Inf. Process.* **2019**, *18*, 1–26. [CrossRef]
112. Lu, C.J.; Aronson, A.R.; Shooshan, S.E.; Demner-Fushman, D. Spell checker for consumer language (CSpell). *J. Am. Med Informatics Assoc. JAMIA* **2019**, *26*, 211–218. [CrossRef] [PubMed]
113. Mammadov, S. Neural Spelling Correction for Azerbaijani Language. In Proceedings of the 13th IEEE International Conference on Application of Information and Communication Technologies, AICT 2019, Baku, Azerbaijan, 23–25 October 2019. [CrossRef]
114. Roy, S. Denoising Sequence-to-Sequence Modeling for Removing Spelling Mistakes. In Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019, Dhaka, Bangladesh, 3–5 May 2019. [CrossRef]
115. Yang, L.; Li, Y.; Wang, J.; Tang, Z. Post text processing of chinese speech recognition based on bidirectional LSTM networks and CRF. *Electronics* **2019**, *8*, 1249. [CrossRef]
116. Zhang, S.; Lei, M.; Yan, Z. Investigation of transformer based spelling correction model for CTC-based end-to-end Mandarin speech recognition. In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, Graz, Austria, 15–19 September 2019; pp. 2180–2184. [CrossRef]
117. Zhou, Y.; Porwal, U.; Konow, R. Spelling correction as a foreign language. In *2019 SIGIR Workshop on eCommerce, eCOM 2019*; CEUR-WS: Aachen, Germany, 2019; Volume 2410.
118. Barteld, F.; Biemann, C.; Zinsmeister, H. Variations on the theme of variation: Dealing with spelling variation for fine-grained POS tagging of historical texts. In Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria, 19–21 September 2018; Austrian Academy of Sciences: Wien, Austria; Institut für Germanistik, Universität Hamburg: Hamburg, Germany, 2018; pp. 202–212.
119. Sooraj, S.; Manjusha, K.; Anand Kumar, M.; Soman, K. Deep learning based spell checker for Malayalam language. *J. Intell. Fuzzy Syst.* **2018**, *34*, 1427–1434. [CrossRef]
120. Sbattella, L.; Tedesco, R. How to simplify human-machine interaction: A text complexity calculator and a smart spelling corrector. In Proceedings of the 4th EAI International Conference on Smart Objects and Technologies for Social Good, GOODTECHS, Bologna, Italy, 28–30 November 2018; pp. 304–305. [CrossRef]
121. Silfverberg, M.; Kauppinen, P.; Lindén, K. Data-Driven Spelling Correction using Weighted Finite-State Methods. In Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata, Berlin, Germany, 12 August 2016; pp. 51–59. [CrossRef]
122. Eger, S.; vor der Brück, T.; Mehler, A. A Comparison of Four Character-Level String-to-String Translation Models for (OCR) Spelling Error Correction. *Prague Bull. Math. Linguist.* **2016**. [CrossRef]
123. Lindén, K. Multilingual modeling of cross-lingual spelling variants. *Inf. Retr.* **2006**. [CrossRef]
124. Abandah, G.A.; Graves, A.; Al-Shagoor, B.; Arabiyat, A.; Jamour, F.; Al-Taee, M. Automatic diacritization of Arabic text using recurrent neural networks. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2015**, *18*, 183–197. [CrossRef]
125. Hasan, S.; Heger, C. Spelling Correction of User Search Queries through Statistical Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), Lisbon, Portugal, 17–21 September 2015; pp. 451–460.
126. Lai, K.K.H.; Topaz, M.; Goss, F.R.F.; Zhou, L.L. Automated misspelling detection and correction in clinical free-text records. *J. Biomed. Informatics* **2015**, *55*, 188–195. [CrossRef]
127. Ramasamy, L.; Rosen, A.; Stranák, P. Improvements to Korektor: A Case Study with Native and Non-Native Czech. In *ITAT (Information technologies–Applications and Theory)*; CEUR-WS: Aachen, Germany, 2015; pp. 73–80.
128. Church, K.W.; Gale, W.A. Probability scoring for spelling correction. *Stat. Comput.* **1991**, *1*, 93–103. [CrossRef]
129. Evershed, J.; Fitch, K. Correcting noisy OCR: context beats confusion. In Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage—DATeCH '14, Madrid, Spain, 19–20 May 2014; ACM: New York, NY, USA, 2014; pp. 45–51. [CrossRef]

130. Makazhanov, A.; Makhambetov, O.; Sabyrgaliyev, I.; Yessenbayev, Z. Spelling correction for Kazakh. In *Computational Linguistics and Intelligent Text Processing, Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2014, Kathmandu, Nepal, 6–12 April2014*; Lecture Notes in Computer Science; Gelbukh, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8404, pp. 533–541. [CrossRef]
131. Mitankin, P.; Gerdjikov, S.; Mihov, S. An Approach to Unsupervised Historical Text Normalisation. In Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage—DATeCH '14, Madrid, Spain, 19–20 May 2014; ACM: New York, NY, USA, 2014; pp. 29–34. [CrossRef]
132. Sariev, A.; Nenchev, V.; Gerdjikov, S.; Mitankin, P.; Ganchev, H.; Mihov, S.; Tinchev, T. Flexible Noisy Text Correction. In Proceedings of the 11th IAPR International Workshop on Document Analysis Systems, DAS 2014, Tours-Loire Valley, France, 7–10 April 2014; pp. 31–35. [CrossRef]
133. Wang, Z.; Xu, G.; Li, H.; Zhang, M. A Probabilistic Approach to String Transformation. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1063–1075. [CrossRef]
134. Huang, Y.; Murphey, Y.L.; Ge, Y. Automotive diagnosis typo correction using domain knowledge and machine learning. In Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013, Singapore, 16–19 April 2013; pp. 267–274. [CrossRef]
135. Reffle, U.; Ringlstetter, C. Unsupervised profiling of OCRed historical documents. *Pattern Recognit.* **2013**, *46*, 1346–1357. [CrossRef]
136. Duan, H.; Li, Y.; Zhai, C.; Roth, D.; Ave, N.G. A discriminative model for query spelling correction with latent structural SVM. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2012; pp. 1511–1521.
137. Rashwan, M.A.A.; Al-Badrashiny, M.A.S.A.A.; Attia, M.; Abdou, S.M.; Rafea, A. A stochastic Arabic diacritizer based on a hybrid of factorized and unfactorized textual features. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *19*, 166–175. [CrossRef]
138. Takasu, A. Bayesian similarity model estimation for approximate recognized text search. In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26–29 July 2009; pp. 611–615. [CrossRef]
139. Takasu, A.; Aihara, K. DVHMM: Variable length text recognition error model. In Proceedings of the 16th International Conference on Pattern Recognition, Quebec City, QC, Canada, 11–15 August 2002; pp. 110–114. [CrossRef]
140. Magdy, W.; Darwish, K. Effect of OCR error correction on Arabic retrieval. *Inf. Retr.* **2008**, *11*, 405–425. [CrossRef]
141. Beaufort, R.; Mancas-Thillou, C. A weighted finite-state framework for correcting errors in natural scene OCR. In Proceedings of the 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, 23–26 September 2007; pp. 889–893. [CrossRef]
142. Byun, J.; Rim, H.C.; Park, S.Y. Automatic spelling correction rule extraction and application for spoken-style Korean text. In Proceedings of the ALPIT 2007 6th International Conference on Advanced Language Processing and Web Information Technology, Luoyang, China, 22–24 August 2007; IEEE Computer Society: Washington, DC, USA, 2007; pp. 195–199. [CrossRef]
143. Magdy, W.; Darwish, K. Arabic OCR error correction using character segment correction, language modeling, and shallow morphology. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing—EMNLP '06, Sydney, Australia, 22–23 July 2006; Association for Computational Linguistics: Morristown, NJ, USA, 2006; pp. 408–414. [CrossRef]
144. Oncina, J.; Sebban, M. Learning stochastic edit distance: Application in handwritten character recognition. *Pattern Recognit.* **2006**, *39*, 1575–1587. [CrossRef]
145. Gong, H.; Li, Y.; Bhat, S.; Viswanath, P. Context-sensitive malicious spelling error correction. In Proceedings of the World Wide Web Conference, WWW 2019, San Francisco, CA, USA, 13–17 May 2019; pp. 2771–2777. [CrossRef]
146. Kulis, B. Metric learning: A survey. *Found. Trends Mach. Learn.* **2012**, *5*, 287–364. [CrossRef]
147. Bellet, A.; Habrard, A.; Sebban, M. A Survey on Metric Learning for Feature Vectors and Structured Data. *arXiv* **2013**, arXiv:1306.6709, 1–59.

148. Needleman, S.B.; Wunsch, C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **1970**, *48*, 443–453. [CrossRef]
149. Bilenko, M.; Mooney, R.J. Adaptive duplicate detection using learnable string similarity measures. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 39–48. [CrossRef]
150. Kim, S.R.; Park, K. A dynamic edit distance table. *J. Discret. Algorithms* **2004**, *2*, 303–312. [CrossRef]
151. Hyyrö, H.; Narisawa, K.; Inenaga, S. Dynamic edit distance table under a general weighted cost function. In *SOFSEM 2010: Theory and Practice of Computer Science, Proceedings of the International Conference on Current Trends in Theory and Practice of Computer Science, Špindleruv Mlýn, Czech Republic, 23–29 January 2010*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 5901 LNCS, pp. 515–527. [CrossRef]
152. Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, Prague, Czech Republic, 23–30 June 2007; Association for Computational Linguistics: Stroudsburg, PA, USA, 2007; pp. 177–180.
153. Gerdjikov, S.; Mitankin, P.; Nenchev, V. Realization of common statistical methods in computational linguistics with functional automata. In Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013, Hissar, Bulgaria, 9–11 September 2013; INCOMA Ltd. Shoumen, BULGARIA and Association for Computational Linguistics: Stroudsburg, PA, USA, 2013; pp. 294–301.
154. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics—ACL '02, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318. [CrossRef]
155. Voorhees, E.M. The TREC-8 Question Answering Track Report. *Nat. Lang. Eng.* **1999**, *7*, 77–82. [CrossRef]
156. Reynaert, M. On OCR ground truths and OCR post-correction gold standards, tools and formats. In *DATeCH 2014: Digital Access to Textual Cultural Heritage 2014, Madrid, Spain, 19–20 May 2014*; ACM: New York, NY, USA, 2014; pp. 159–166. [CrossRef]
157. Lueck, G. A data-driven approach for correcting search queries. In Proceedings of the Spelling Alteration for Web Search Workshop, Bellevue, WA, USA, 19 July 2011; p. 6.
158. Tseng, Y.H.; Lee, L.H.; Chang, L.P.; Chen, H.H. Introduction to SIGHAN 2015 Bake-off for Chinese Spelling Check. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, Beijing, China, 30–31 July 2015; Association for Computational Linguistics: Stroudsburg, PA, USA; pp. 32–37. [CrossRef]
159. Wu, S.H.; Liu, C.L.; Lee, L.H. Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013. In Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, Nagoya, Japan, 14–18 October 2013; Asian Federation of Natural Language Processing: Nagoya, Japan, 2013; pp. 35–42.
160. Sorokin, A.; Baytin, A.; Galinskaya, I.; Rykunova, E.; Shavrina, T. SpellRuEval: The first competition on automatic spelling correction for Russian. In Proceedings of the International Conference "Dialogue 2016", Moscow, Russia, 1–4 June 2016.