Literature review

In this chapter we will take a look at the existing spellcheckers developed using a data driven statistical language model. We investigate the types of errors and the way this spellcheckers detect and correct errors. Spelling error detection and correction techniques are designed on the basis of different natures of errors. Studies were previously performed to analyze various spelling error types. [Jurafsky el.at, 1992]Some authors split spelling errors into non-word errors, isolated-word errors and real-word errors. These errors occur as typos or when the pronunciations of a misspelled word are assumed to be of an intended correct word.

Spelling errors

There are two main type of errors most spellcheckers aim at Non-word errors and real-word errors. Non-words errors are spelling errors resulting in words that do not appear in the reference dictionary and real-word errors are words that are in the reference dictionary but are actually erroneous spellings of some other words [Tavast, el at, 2012]. A spelling checker will detect a misspelled word and depending on the level error, fine tune the word to provide a set of suggestions. These suggestions are a set of words a user probably intended to type. Non-word errors are relatively easier to detect and eradicate. Real word errors are more intricate ones. Usually, such error affects the syntax and semantics of the whole sentence, which in some cases requires human-being involvement for detection [Cheng, el at, 2007]. The use of spelling correctors should be handled with care. This is because some errors are attributed by auto-correctors, a feature of some word processing software [Hirst, el at, 2005]. A person may input a word they are not sure of and an auto-corrector can give a completely wrongly. Especially if that person is foreign to the language in use.

Error Detection

The detector module is responsible for determining if a word is considered misspelled or not. N-gram analysis can be a detector.

N-gram analysis

An n-gram is an n letter subsequence of a string, where n usually is 1, 2, or 3. N-grams can be unigrams for one letter, bigram for two letters and trigram for three letters respectively. In general, n-gram analysis techniques check each n-gram in an input and compare it against an existing table of n-gram statistics. Spellcheckers using N-gram analysis makes use of N-gram statistics in a text corpus [Wasala el.at, 2010]. N-gram statistics is the frequency counts or probability of occurrence of N-grams. N-gram analysis is used to determine correctness of words in a mass of text [Gupta, el.at, 2012]. N-gram based techniques used without the dictionary can be used to find the position at which in the misspelled word an error occurred. This is in some cases achieved by employing character-based N-gram language models [WU, el.at, 2013] but can also use word-based N-gram language models. N-gram analysis is a statistical approach and statistical approaches are mainly influenced by corpora, its size and correctness. Lexical diversity is also an essential factor to access in a corpus. The frequency statistics for each word is compared with the system threshold. This threshold divers from one spellchecker to the other

[Bidyut, el at]. Normally, if the frequency is below the threshold the word is identified as wrongly spelled. This means n-grams that do not occur or with infrequency occurrences are considered to be misspellings.

Error correction

The corrector module is responsible for providing a set of possible corrections for a misspelled word. After a word is flagged as wrongly spelled, if possible a set of suggestion is availed. Studies point that most misspellings involve at most one character change from the intended word [Kukich, el at, 1992]. This means the distance between the correct word and the misspelled word is ±1 character difference. Accordingly to some authors [Cheng, el at, 2007], n-grams can also be used for error correction. This is done by assuming certain n-grams within a word are correctly spelled and fix the remaining n-grams. A list of words is established as suggestions. It is also important to rank the words and lift the closest suggestion to the top of the list and presumably trim it. To organize this we need an algorithm that computes the minimum edit distance. A Levenshtein distance can accomplish this. It will suffice as it will show the shortest distance between suggested words and the word with the shortest distance will be considered as the best suggestion.

Data

This chapter investigates data driven statistical model. Therefore, it is important to also discuss the data component of the model and see how it is built.

Corpus

A corpus is a large collection of texts [Robin, 2007]. A spellchecker is as good as its corpus. Therefore, we need to be careful in selecting and using corpora. Availability of such corpora is of the essence as well. There already exists an open source morphological isiZulu corpus, Ukwebelana [Spiegler, el.at, 2010]. The corpus uses actual language which enables future research and use. Statistical approaches are mainly influenced by corpora, its size and correctness. A corpus can be built from collected documents in public archives, libraries, consulting language experts and last but not least using already existing dictionaries. Language evolves so should the corpora to yield expected results. Outdated corpora means currently introduced words to the language will not be recognized.

Conclusions

Spelling correction must involve a dictionary or corpus, since the set of possible correct spellings are defined in terms of membership in the chosen corpus. IsiZulu consists of two components lexicon and mythological analyzer which can be extended using a language corpora. The issue is not only construction of corpus but also optimization of its access. In the project statistical analysis of a corpus information improved. E.g. considering limitations of a frequency table. And it should also be in a position to recognize a Zulu word written as one word and when split provided the pieces are successive. In the light of the review above, we conclude that a data

driven statistical language model spellchecker needs data to decide whether a word is correctly spelled or not. Data is core for this spellchecher.

References

Daniel Jurafsky, James H. Martin, *Speech and Language Processing*, Pearson. 1992

Asanka Wasala, Ruvan Weerasinghe, Randil Pushpananda, Chamila Liyanage & Eranga Jayalatharachchi. A Data-Driven Approach to Checking and Correcting Spelling Errors in Sinhala. *The International Journal on Advances in ICT for Emerging Regions*. 2010 03 (01) : 11 – 24

Neha gupta, Pratistha mathur Spell checking techniques in NLP: a survey. *Department of computer science, Banasthali vidyapith,India* . 2012

Jian-cheng Wu, Hsun-wen Chiu, & Jason S. Chang. Integrating Dictionary and Web N-grams for Chinese Spell Checking. *Computational Linguistics and Chinese Language Processing* Vol. 18, No. 4, December 2013, pp. 17-30

S. Spiegler, A. van der Spuy and P. A. Flach. Ukwabelana - An Open-source Morphological Zulu Corpus. Proceedings of the 23rd International Conference on Computational Linguistics (COLING, 2010), 1020–1028
Robin. Natural language processing. Article on natural language processing. Dec 8th, 2009

Arvi Tavast, Maire Koit & Kadri Muischnek. 2012. Human Language Technologies: The Baltic Perspective. *Proceedings of the Fifth International Conference Baltic HLT 2012.* IOS Pres BV. Netherlands

G. Hirst and A. Budanitsky. Correcting real-word spelling errors by restoring lexical cohesion. Natural Language Engineering, 11(1):87–111, March 2005

Cheng, Charibeth, Alberto, Cedric Paul, Chan, Ian Anthony and Querol, Vazir Joshua. SpellCheF : Spelling Checker and Corrector for Filipino. *journal of research in science, computing, and engineering.* Vol. 4 No. 3 December 2007.

Bidyut B. Chaudhuri. A simple real-word error detection and correction using local word bigram and trigram. *Proceedings of the Twenty-Fifth Conference on Computational Linguistics and Speech Processing (ROCLING 201*3).

Kukich, K. Techniques for Automatically Correcting Words in Text. *ACM Computing Surveys*. Volume 24, Number 4. pp. 377-439. December 1992.