Lexical Normalisation taking into account one-to-multiple relations

There are already automatically generated corpora that deal with lexical normalization. For example the manually annotated corpora lexnorm [3] and a manually annotated sms corpus [1]. But also larger automatically generated corpora are available, for example the emnlp corpus [2], and a sms corpus [1] generated using the manually annotated corpus. When using these corpora, their weakness becomes visible quickly: they cannot deal with multiple token expressions. See some example entries of the emnlp corpus below:

```
annny any
spcial special
yesterd yesterday

ure your
iwilll ill
```

The first three examples are all correct, but the 4th and 5th example are wrong. These are the 2 main categories of errors, the first one is an intended error while the second one is a typo. The out of vocabulary (OOV) tokens can impossibly be replaced by 1 in vocabulary (IV) token without losing meaning.

My thesis proposal is to automatically recognize when a token should be replaced by two (or more) tokens. If this system is finished a large corpus like emnlp can also be generated, by simply running the system to a lot of raw data. There are multiple approaches possible to achieve this goal, but I only include some basic ideas in the rest proposal.

The token can be split in all possible places (or maybe less), and then perhaps the separate words become visible. For typos like 'iwill' this is easy:

```
i will
iw ill
iwi ll
iwil l
```

Only the first line contains two IV tokens, so this is the correct replacement. For internet slang this approach is probably less robust:

```
u re  ->  you are
ur e
```

The problem is at the arrow, single word recognition is necessary here. This is not a new problem though, and existing approaches like the edit distance (for the 2nd word) and the double metaphone algorithm [4] (for the 1st word) can be used for this.

Another approach is to use the context. Context can provide us with the most probable words that should fit in the gap. This probably works best when combined with the previous approach. Because with only the context, the target tokens might be very different then the source token.

Downloads

- The manually annotated lexnorm:
  `http://www.csse.unimelb.edu.au/research/lt/resources/lexnorm/`
- Large automatically generated emnlp:
  `www.cs.mu.oz.au/~hanb/emnlp.tgz`
- The sms corpus is not publically available for as far as I know.

References

[1] Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. Investigation and modeling of the structure of texting language. *International Journal of Document Analysis and Recognition (IJDAR)*, 10(3-4):157–174, 2007.

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

[3] Bo Han, Paul Cook, and Timothy Baldwin. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):5, 2013.

[4] Lawrence Philips. The double metaphone search algorithm. *C/C++ users journal*, 18(6):38–43, 2000.