Unsupervised Morphological Paradigms for Spelling Check

Anonymous NAACL submission

Abstract

Grouping of various morphosyntactically similar words could reveal consistent patterns in the way the words are inflected in a language. In this paper, we propose a method that takes limited knowledge of the language and finds such similar groups. Groups are based purely on inflectional forms of words. Unsupervised method of learning is adopted to cluster using the features extracted from the words. The results obtained are quite natural to human interpretation. Restricting our scope to verbs, transformation rules for obtaining the inflected forms of words are derived. These rules can be used to detect the spelling errors in various inflected words of the language.

1 Introduction

In morphological paradigms, we find a set of related forms, usually *inflected forms* of the word with a given lexeme. For example, a verb *sail* has *sails* (‘PRESENT,3SG’), *sailing* (‘GERUND’) and *sailed* (‘PAST’) as inflected forms in its linguistic paradigms. Morphological paradigms can be used to obtain inflected forms of the words from the lexeme, and vice versa.

Durrett and DeNero (2013) proposed a supervised approach towards prediction of inflected forms of the given lexical form of a word. Their system detects orthographic transformation rules from the supervised inflection table data from English Wiktionary, and then learns the n-gram context features. Lee (2015) explored the computational structure of morphological paradigms from the perspective of unsupervised learning. The three main focus areas of his work involved: (i) stem identification, (ii) paradigmatic similarity, and (iii) paradigm induction.

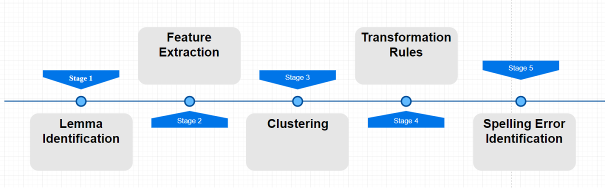

Figure 1: Overall process

1.1 Goal

In this paper, our approach is to augment previous work and employ an unsupervised method to (i) obtain lexeme from morphological paradigms, (ii) extract the features for distinguishing between various lemma and their supposedly inflected forms, (iii) group lemma which inflect in similar manner, (iv) derive the transformation rules, and (v) apply these rules to identify spelling errors, in which we consider as the errors in inflectional forms of the words. In summary, we learn morphological paradigms using unsupervised learning and acquire the orthographic transformation rules. We test these rules for spelling check. This paper consists of as follows: we describe the overall process in detail (Section §2), show results of experiments (Section §3), and discuss the limit of the proposed method and a conclusion (Sections §4 & §5).

2 Overall Process

In this section, we describe our unsupervised process for identifying the spelling errors in inflected forms of words. We use five major steps to describe the process as shown in Figure 1.

2.1 Lemma identification

Lexeme is the canonical form of the word in a natural language. However, for a completely unsupervised scenario, i.e., with no prior information regarding the language or its formation, we define the pseudo-lexeme to as the largest common sub-string from the word and its supposedly inflected forms. We extract the lemma from the list of its derived forms. For example, for words *combine, combined, combines, combining*, the pseudo-lexeme would be *combin* as it is the longest sub-string which appears in all the forms of the word. Also, irregular words, which do not follow any specific pattern in inflated forms such as *go-went* or *come-came* are eliminated in this step.

2.2 Feature extraction

Deciding on the features to use for grouping words that inflect in similar ways is the core part of the process. The concatenating suffixes of the words distinguish the words from each other. This, the suffixes are identified as features for defining the uniqueness of that particular lemma form. In the example considered above, features of pseudo-lexeme *combin* would be *ed, es, ing* which on concatenating with lemma give the inflection *combined-combines-combining*. Using the separated lemma and suffixes, the feature matrix is formed. The feature matrix stores the information about all the lemma and the suffixes that are introduced to form derived forms. This matrix is used as an input in the next step for the clustering algorithm.

2.3 Clustering

Lemma having similar feature patterns can be clustered together using hierarchical clustering. The agglomerative clustering algorithm is used to form the clusters. The number of clusters are altered as per the general human interpretation of a language. The intuition regarding words known to inflect similarly is used to decide the ideal number of different groupings required for a language. We set 10 as number of clusters required, which gives an accurate result for English. Some examples of words that are grouped together in a cluster are given in Table 1. The results are natural to human interpretation. For example, in cluster 0, the derived forms of the words are same as original lemma form (*cut*). Or in cluster 8, words ending with *do* are grouped together; and last *o* is replaced by suffix *id, ing, oes* to generate the inflated forms.

2.4 Transformation rules

Various inflectional forms are aligned based on the logic of longest common sub-strings among the given forms of the pseudo-lexeme. This gives a more efficient linear method towards alignment of various forms of lemma than the one proposed by Durrett and DeNero (2013) with the polynomial time because of the minimum edit distance. This alignment gives the transformation rules for each cluster, which are then generalized. A set of suffixes that defines the uniqueness of each cluster would be obtained. These transformation rules can be used to get the assumed correct inflected forms for a pseudo-lexeme. In CELEX English dataset, the information about the inflected forms of the verbs are given, but any information regarding the form of the verb (person and tense) is not used. As a consequence, the derived rules that simply indicate what changes should be made to the pseudo-lemma are general, they are not person or tense specific. The rules we extracted for the above clusters are as follows (partially presented):

cluster	examples
0	offset, cut
1	burst, learn, burn
2	reline, decide, stone
3	nod, knit, rub
4	pay, unsay, prepay
5	sail, snowball, recommend
6	rebuild, spend, lend
7	cry, fancy, bury
8	do, undo, redo
9	snitch, hiss, fox

Table 1: Cluster representatives

- Cluster 3: (*snog, stop*)

$$\begin{aligned}\text{Rule}(\epsilon) &= \text{C}ed\\ \text{Rule}(\epsilon) &= s\\ \text{Rule}(\epsilon) &= \text{C}ing\end{aligned}$$

- Cluster 5: (*sail, reconsider*)

$$\begin{aligned}\text{Rule}(\epsilon) &= ed\\ \text{Rule}(\epsilon) &= s\\ \text{Rule}(\epsilon) &= ing\end{aligned}$$

```
Correct word : besiege
Correct word : gaze
Incorrectly Inflected word : removieng
Correct word : peruse
Incorrectly Inflected word : scord
Correct word : care
Correct word : accuse
Correct word : instance
...
```

Figure 2: Sample output

- Cluster 7: (*declassif*)

$$\text{Rule}(\epsilon) = ied$$
$$\text{Rule}(\epsilon) = ies$$
$$\text{Rule}(\epsilon) = ying$$

where C represents the last consonant of lemma which gets repeated in inflated form. Here we are considering the entire root verb as lemma and simply adding the suffixes *ed, s, ing* in order to form derived words. Hence using presented rules, ϵ (null) is replaced by the suffixes: *declassif* + *ied*|*ies*|*ying* = *declassified*|*declassifies*|*declassifying.*

2.5 Spelling error identification

The above described process is repeated for unseen test data. First, we identify the pseudo-lemma, then extract the suffixes based on which the target cluster could be identified by the clustering algorithm. The algorithm classifies the test pseudo-lemma with the most similar pseudo-lemma from the training data based on the inflectional suffixes both have in common. The hence identified cluster is then used to decide the transformation rules that can be applied to the unseen testing data to generate the supposedly correct inflected forms of the test pseudo-lemma. The application of those transformation rules gives the supposedly correct inflected forms of the test pseudo-lemma, which are then compared with the current test word in question. Inconsistencies in the way the words are inflected can hence be identified. Figure 2 shows the sample output where the system detect incorrectly inflected words such as *removieng* ('removing') or *scord* ('scored').

3 Experiments

3.1 Results

A total of 17,848 inflected forms of the verb are used for experimentation. They are all from the CELEX lexical databases of English (Version 2.5)[1]. We identify 4,462 pseudo-lemma as shown in Figure 3. The feature matrix which are about the pseudo-lemma (line) and the suffixes (column) is obtained as in Figure 4. Then, the clustering algorithm is performed and transformation rules are extracted. Examples of transform rules are presented in §2.4 as well as a clustering result already shown in Table 1.

3.2 Identifying spelling errors

For evaluation purposes of spelling error check, several erroneous words were arbitrarily inflected to get incorrect forms (spelling errors). For instance, a typographical mistake could be because of a wrong sequence of letters as in *remvieng* instead of *removing*, or it could be a miss on a letter from the word, in case of *scord* as correctly for '*scored*'. These incorrect inflectional forms of the words are identified by finding the cluster where the word could be most similar to. That cluster's inflections are appended to the current test pseudo-lemma. If the obtained inflectional forms are different than the test inflectional form of the word, it is identified as a spelling or inflectional error for that language as shown in Figure 2.

4 Discussion

A couple of limitations have been identified for the proposed approach. We describe them in this section including previous work in which we did not explore their methods in this paper.

4.1 Non-concatenative morphology

In Arabic, *k-t-b* is the lexeme in *kataba* ('he wrote') or *kattaba* ('he caused to write') (McCarthy, 1981). For such a morphological construction, while the minimum edit distance proposed in Durrett and DeNero (2013) can deal with this kind of morphology, our approach fails because of its linearity characteristics. Since our method is completely unsupervised, non-concatenative morphology would be difficult to be implemented. To employ morphological

[1] https://catalog.ldc.upenn.edu/LDC96L14

```
'eat': ['ate', 'eat', 'eaten', 'eating', 'eats'],
'subserve': ['subserve', 'subserved', 'subserves', 'subserving'],
'sail': ['sail', 'sailed', 'sailing', 'sails'],
'snuffle': ['snuffle', 'snuffled', 'snuffles', 'snuffling'],
'recount': ['recount', 'recounted', 'recounting', 'recounts'],
'snuff': ['snuff', 'snuffed', 'snuffing', 'snuffs'],
'unveil': ['unveil', 'unveiled', 'unveiling', 'unveils'],
'perplex': ['perplex', 'perplexed', 'perplexes', 'perplexing'],
'legalize': ['legalize', 'legalized', 'legalizes', 'legalizing'],
...
```

Figure 3: Results on pseudo-lemma identification

	d	ed	s	Ced	...
extend	0	1	1	0	
affect	0	1	1	0	
achieve	1	0	1	0	
nod	0	0	1	1	
	...				

Figure 4: Feature matrix

paradigms, either the language has to be concatenative for unsupervised learning, or the approach should be supervised. Otherwise there are no means available to identify the lexeme from a list of inflected forms.

4.2 Irregular words

This approach works on concatenative words for which suffixes are appended to a part of root verb to form the derived form. However, there is a class of words such as *go-went, eat-ate, catch-caught* which inflate in a regular manner. For such words, a different fashion for clustering will have to be considered. Currently, the features are identified by finding the longest common sub-string among the various supposedly correct inflected forms of that particular pseudo-lemma. However, we can handle the case of irregular words by finding features across different inflectional forms of words. For instance, pattern among *taught, fought, caught* can provide substantial information for grouping the respective pseudo-lemma *teach, fight, catch* together.

4.3 Spelling error checking

While spelling errors have been explored frequently in the learner corpus, for example the NUS Corpus of Learner English (Dahlmeier et al., 2013), our approach for spelling check presented in this paper is based on a typographical error. Dealing with real-world spelling errors is mostly relied on grammaticality on the well-formedness of words. They are mostly grammar related errors (Granger, 2003) which are beyond the scope of this paper.

4.4 Previous work

We can also consider two previous approaches: unsupervised as in Dreyer and Eisner (2011) with a Dirichlet process mixture model to discover morphological paradigms and predict the spellings of unobserved forms, and semi-supervised by Hulden et al. (2014) where they generalized a few manually specified inflection tables into an abstract form for morphological paradigms.

5 Conclusion

This paper presented unsupervised morphological paradigms, and being inspired by transformation rules of the supervised method, we experimented the spelling check for inflected words.

References

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Markus Dreyer and Jason Eisner. 2011. Discovering Morphological Paradigms from Plain Text Using a Dirichlet Process Mixture Model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 616–627, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Greg Durrett and John DeNero. 2013. Supervised Learning of Complete Morphological Paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195, Atlanta, Georgia. Association for Computational Linguistics.

Sylviane Granger. 2003. The international corpus of learner English: a new resource for foreign language learning and teaching and second language acquisition research. *TESOL Quarterly*, 37(3):538–546.

Mans Hulden, Markus Forsberg, and Malin Ahlberg. 2014. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578, Gothenburg, Sweden. Association for Computational Linguistics.

Jackson Lee. 2015. Morphological Paradigms: Computational Structure and Unsupervised Learning. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 161–167, Denver, Colorado. Association for Computational Linguistics.

John J. McCarthy. 1981. A Prosodic Theory of Nonconcatenative Morphology. *Linguistic Inquiry*, 12(3):373–418.