Inflection Algorithms

Akshay Srivatsan, Ian Mukherjee, Nathan Smith

April 18, 2016

1 Introduction

For this assignment, our group incrementally implemented inflectors incorporating lemma bigrams, part-of-speech tags, and dependency trees. These separate inflectors were then combined and a backoff structure was used to determine the optimal morphological choice.

2 Bigram Model:

A bigram inflector was first implemented, where the sentence's previous lemma is also considered. This was done by adding new dictionary keys represented by tuples of the current and previous lemma, with $< s >$ substituting for the first lemma in the sentence. When testing, if the test sentence's lemma bigram does not have a key, the current lambda is chosen by backoff.

The bigram model yielded the following results on dtest:

Accuracy for Bigram Model:

Accuracy	0.60

3 Part-of-Speech Model:

We then built an inflector that considered a word's part-of-speech. This program follows the same structure, with the keys now represented by tuples of the current lemma and the word's part of speech. During testing if the lemma POS tuple was not built during training, the program defaults to the best inflection for the lemma key.

The POS model yielded the following results on dtest:

Accuracy	0.66

4 Dependency Tree Model:

Next, we built and inflector that considers the word's dependency tree. In the dependency tree files, each word is a node in the tree corresponding to the sentence the word belong to. The parent and label information from the dependency tree is stored as a typle $(parent_index, label)$, and the tuple $(lemma, tree)$ is used as a new dictionary key.

The dependency tree model yielded the following results on dtest:

Accuracy	0.64

5 Tree + POS Model:

We then combined the dependency tree and part of speech models. A new dictionary key holding the lemma, tree tuple, and part of speech was created. This $(lemma, pos, tree)$ key was given precedence, followed by $(lemma, pos)$, $(lemma, tree)$, and $(lemma)$ in that order, reflecting the previous accuracy results we obtained for each individual model.

The Tree + POS Model yielded the following results on dtest:

Accuracy	0.67

6 BPT Model:

Finally, we combined the bigram, part of speech, and dependency tree metrics into a single inflector. The eight keys were given the following order of precedence:

1. (prev-lemma, lemma, tag, tree)
2. (prev-lemma, lemma, tag)
3. (prev-lemma, lemma, tree)
4. (lemma, tag, tree)
5. (lemma, tag)
6. (lemma, tree)

7. (prev-lemma, lemma)
8. (lemma)

During testing, this sequence of keys was used for backoff, again following the previous accuracies calculated for each of the individual models that made up these keys.

The BPT Model yielded the following results on dtest:

Accuracy	0.69

7 Bibliography:

References

[1] Mark Hopkins, Jonathan May. Tuning as ranking.

[2] Nadir Durrani, Helmut Schmid, Alexander Fraser, Philipp Koehn, Hinrich Schutze The operation Sequence Model

[3] Daniel Ortiz Martinez, Ismael Garcia Varea, Francisco Casacuberta Nolla Generalized Stack Decoding Algorithms for Statistics Machine Translation