

Machine Translation Inflection

Haitang Hu
hthu@cs.jhu.edu

April 27, 2015

1 Methods

1.1 Bigram Language Model

Here we simply explore the bigram language model over lemma-form pair. Note that we should also consider the back-off scheme, that if we didn't match the bigram directly, we should reduce and back-off to unigram model, for our first word in the bigram matching. The back-off bigram matching shows a significant improvement over unigram model.

1.2 Bigram with Part-of-Speech

In this model, we also incorporate the Part-of-Speech information onto the bigram model. We now decide a lemma by first looking into its bigram tags and form pairs, so that we are now conditioning POS tags additionally. Again, we also use a back-off scheme here, we first try to match the bigram with tags, then unigram with tags, and finally solely unigram.

This improves bigram language model.

1.3 Bigram with Dependency Tree

Similarly, a dependency tree can be parsed to use as a bigram. Here, we consider the node and its parent in the dependency to be a bigram pair, thus allow us to encode longer bigram dependency instead of word level bigram. For better unigram prediction, we also add Part-of-Speech tag on unigrams. A similar back-off scheme has also been used here : we first match dependency bigram, then unigram with tag, and finally unigram.

This improves a little on bigram with Part-of-Speech.

2 Experiment

Here, we explore the performance of all three above methods, with different N -gram models.

3 Conclusion

As we could see here, incorporating more information in our n -gram model helps us on evaluating the inflection. Also, since we have developed several methods, we could also

Method	Score
bigram	$45817/70974 = 0.65$
trigram	$46194/70974 = 0.65$
4-gram	$46216/70974 = 0.65$
bigram + POS	$48325/70974 = 0.68$
trigram + POS	$48325/70974 = 0.68$
bigram + POS + DEP	$48425/70974 = 0.68$

Table 1: Result

consider to add a voting to use the most possible result that are common in major methods. But due to the time limits, we can't implement and test on the *Ensemble* method.