Homework 5: Inflection

Roger Que

2014–04–29

This submission implements an inflector using bigram probabilities with Katz backoff to determine the best inflected form for each lemma, based on NLTK's *n*-gram model framework [1]. Two types of bigrams are used:

- Bigrams that decompose conventionally from left to right in sentence order, using both the lemma and the inflected form of the previous word as context.
- Bigrams that decompose over the dependency tree, using the lemma of each word's parent node as its context.

The probabilities of each, which are determined from the training data, are combined with a weighted sum to give the total probability of each inflected form, and the one with the highest probability is chosen.

The relative weights of the two bigram features were tuned on the development data. Here, the notation λ is used to represent the weight of the dependency bigram feature, implying a corresponding conventional bigram feature weight of $1 - \lambda$. Values for λ from 0.0 to 1.0, in increments of 0.1, were tested. This process yielded an optimal λ^* of 0.2, which also achieved the highest performance on the testing set.

Development and testing accuracy were strongly correlated across different values of λ, being within one percentage point of each other in all cases. For $\lambda = \lambda^*$, 64% of judgments were correct; for $\lambda = 0.0$, using only conventional bigrams, 63%; and for $\lambda = 1.0$, using only dependency tree bigrams, 62%. From the data, it is clear that the dependency parse information provides a marginal aid in identification of the correct inflected form, but that in isolation it cannot discriminate between forms as well as conventional bigram information. Closer investigation of the results on development data show a high overlap in correct judgments between the $\lambda = 0.0$ and $\lambda = 1.0$ settings, as shown in Table 1.

The choice to use both the lemma and the inflected form of the previous word in the "conventional" bigram probabilities was motivated by ease of implementation, and not by any linguistic factors. The effect of this decision on performance ultimately depends on the presence or absence of a dependence on the previous inflected form, which was not tested. Should the dependence be merely on the lemma and not the inflected form, some incorrect judgments may have been rendered due to the splitting of the bigram probability mass across many different inflected forms.

Further potential avenues for improvement include the use of a longer context for the existing models, as well as the incorporation of additional data from the dependency parse, such as edge labels, into the feature space.

References

[1] Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

	0.0 C		**0.0 I**		**Total**	
1.0 C	40,926	58%	3,160	4%	44,086	62%
1.0 I	3,692	5%	23,196	33%	26,888	38%
Total	44,618	63%	26,356	37%	70,974	100%

Table 1: A comparison of the percentage of correct (C) and incorrect (I) judgments on development data for λ values of 0.0 (conventional bigram features only) and 1.0 (dependency bigram features only).