LING 570 – HW4

Q3

Based on the input lexicon and the morphotactic rules, expand_fsm1.sh creates an expanded FSM in the following form:

(q0 (q1 “a” “a”))
(q1 (q2 “t” “t”))
(q2 (q3 “e” “e”))

The output represents information processed from the lexicon and the morphotactic rules. Each word provided in the input lexicon is broken down into individual characters while the states represented in the expanded FSM is based on the transition states observed from the input morphotactic rules. In addition, the FSM is created only if each class label from the morphotactic rules matches against those that are defined in the lexicon.

The expand_fsm2.sh follows the same processing requirement except that the output FSM also contains the class label that the word is being associated to and is represented in the following form:

(q0 (q1 “a” “a”))
(q1 (q2 “t” “t”))
(q2 (q3 “e” “e/irreg_past_verb_form”))

The addition of class labels in the FSM created from expand_fsm2.sh facilitate the post-processing requirements of the morph_acceptor2.sh, which would check each given word for its composition of stem/class label and suffix/class label.

Q5

The Porter Stemmer is a rule based system that removes suffix from word so that words can be reduced into common terms or groupings. In an information retrieval system, the application of Porter Stemmer reduces the size and complexity of search data and thus delivering the outcome of a faster information retrieval performance. For a given language, the main components of the Porter Stemmer are a list of suffixes and a set of rules that govern each suffix removal. Additionally, the Porter Stemmer does not require a Lexicon to function.

On the other hand, the morphological analyzer comprises of Lexicon, Morphotactics and Orthographic rules. The Lexicon contains a list of stems and affixes of a given language where each stem or affix is being associated to a classification label. The Morphotactics contains the definitions of composing and ordering the classification labels for that language and the Orthographic rules deal with exception handling using language specific linguistic rules. With these three components, the Morphological analyzer could handle more than just suffix removal and may be extended to handle tasks that require more linguistic accuracy.

The functionality of the Porter Stemmer is such that the rules for suffix removal are based on predefined conditions and these are represented in the form of:

(condition) S1 => S2

The condition above consists of the input word and any conditions that the word would have to observe. S1 is the given suffix and if it matches the suffix of the input word, then S1 would be replaced by S2.

In contrast, the functionality of the Morphological analyzer is more extensive. First, in addition to just suffixes each lexical entry in the Lexicon component may contains all possible affixes and stems with their respective class labels. For example:

Stems	Class Label
fish	N
dog	N

Affix	Class Label
s	PL

Second, the Morphotactics component defines how the class labels may be composed or ordered. For example:

PL follows a N

Therefore if a given word is “dogs”, the analyzer is able to evaluate it to be:

dogs => N + P, dog => N + s => PL

Third, as part of the Morphological analyzer the Orthographic rules component offer an additional level of linguistic refinement by handling exception through the use of rules so that the plural of “fish” for example is evaluated to be “fishes” instead of “fishs”

In terms of rules application, the Porter Stemmer executes the predefined rules based on a sequence of ordered steps such that each word is systematically reduced into the final stem. The rules in the Morphological analyzer need not be ordered and can be selectively executed based on the Morphotactics, the lexical entries (stem/affix and class label classification) as defined in the Lexicon and the Orthographic rules. In the Morphological analyzer, addition or removal of rules, or editing lexical entries in the Lexicon may be performed independent of the system. In contrast, similar

operations performed on the Porter Stemmer would mean changing the system's code and their implementation.

From the perspective of system portability to support new language, the Porter Stemmer implementation requires an in-depth knowledge of the new language and the component morpheme structure since these are directly implemented in the software code. While the Morphological analyzer is a system of a higher complexity as compared to the Porter Stemmer, it supports portability across different languages better. Due to its modular characteristic, the three components of the Morphological analyzer can be developed separately beforehand by linguist and once the three components are available, automation would be straight forward given that the codes may be implemented without having to understand the language details.

We can see that the implementation of Porter Stemmer to cover the entire vocabulary of a given language requires lesser effort as compared to the Morphological analyzer. This is because the Porter Stemmer need not consider the full extent of words in the language as long as the rules and suffixes are well defined. Thus, the direct approach of not having to reference a Lexicon and using lesser data for processing gives the Porter Stemmer the advantage over the Morphological analyzer in terms of speed of execution.

In the case of the Morphological analyzer, implementation to cover the whole vocabulary of a given language requires all word stems and affixes to be documented in the Lexicon. Creating a complete lexical entries in the Lexicon, coupled with the need to define the Morphotactics component and Orthographic rules require extensive effort. However, the comprehensive data set along with their classifications would directly translate into the Morphological analyzer yielding better refined and more accurate outputs over the Porter Stemmer.

End of HW4 – submitted by Wee Teck Tan

Student ID: 0937003

Course Name: LING 570