Development of a German-English Translator

Felix Zhang
TJHSST Computer Systems Lab 2007-2008

Abstract

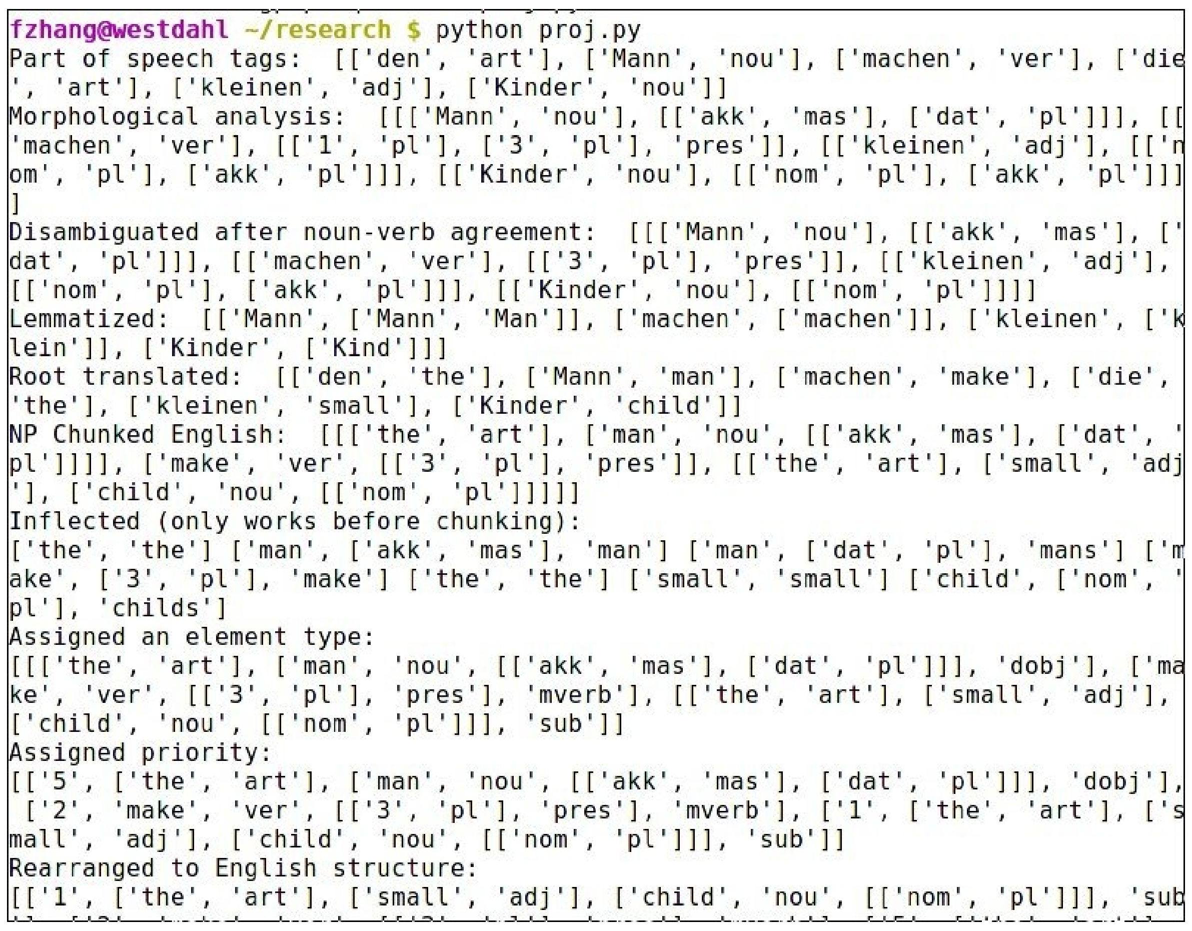
Machine language translation as it stands today relies primarily on rule-based methods, which use a direct dictionary translation and at best attempts to rearrange the words in a sentence to follow the translation language's grammar rules to allow for better parsing on the part of the user. This project seeks to implement a rule-based translation from German to English, for users who are only fluent in one of the languages. For more flexibility, the program will implement limited statistical techniques to determine part of speech and morphological information.

Background

Rule-based translation is the oldest form of language processing. A bilingual dictionary is required for word-for-word lookup, and grammar rules for both the original and target language must be hardcoded in to structure the output sentence and create a grammatical translation. Most online translators currently are based off of rule-based translation systems. Statistical machine translation is based off of a bilingual corpus, which the program uses to "learn" the language. It is much more flexible, being language-independent, but much harder to implement.

Development

The main components to a rule-based translator are a bilingual dictionary, a part of speech tagger, a morphological analyzer that can identify linguistic properties of words, a lemmatizer to break a word down to its root, a method for noun-verb agreement, an inflection tool, and a parse tree. Statistical part-of-speech tagging is implemented with a large German word corpus, with a part of speech assigned to each word. The program determines the most likely tag by checking the frequency of each tag's occurrence.

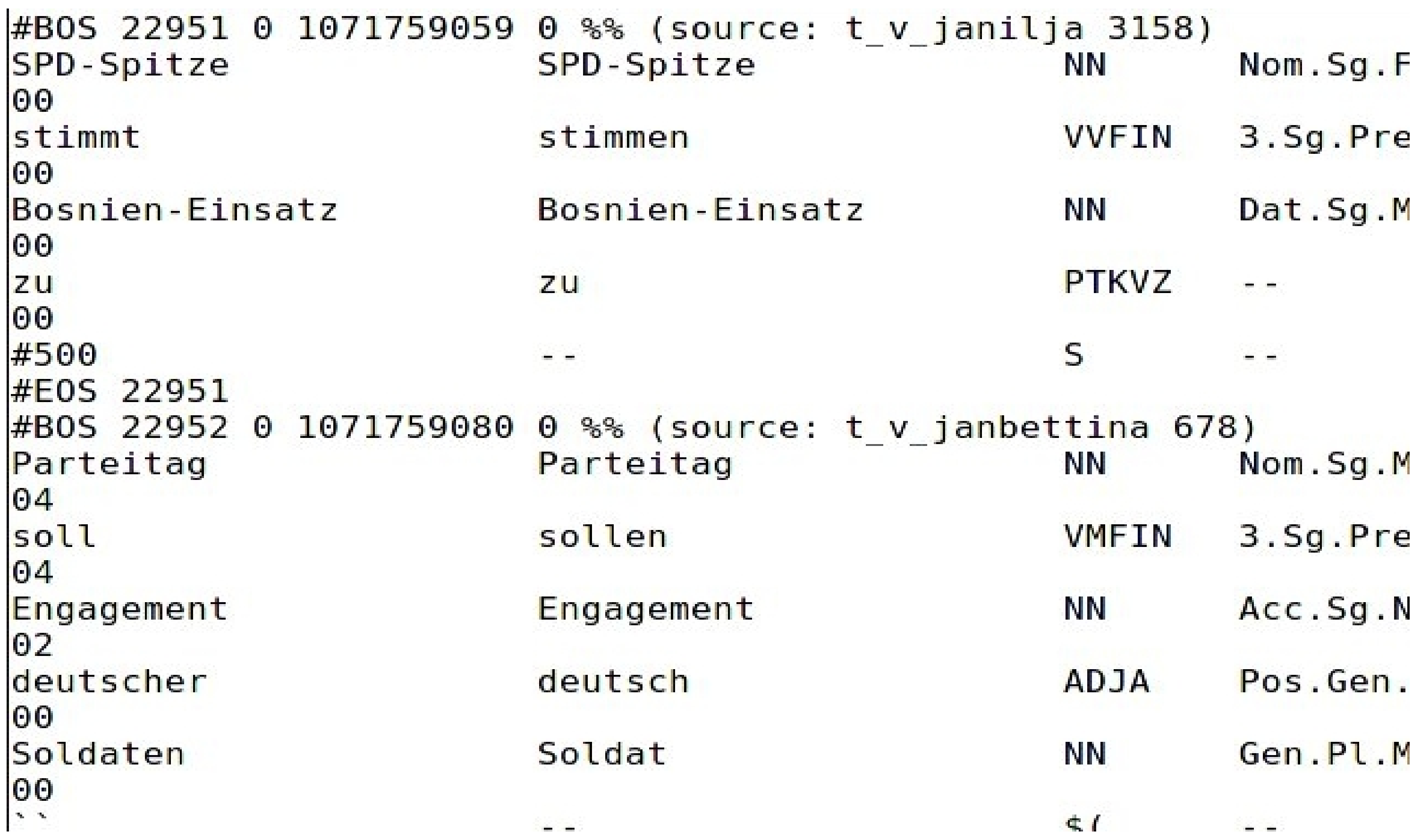


```
fzhang@westdahl ~/research $ python proj.py
Part of speech tags:  [['den', 'art'], ['Mann', 'nou'], ['machen', 'ver'], ['die
', 'art'], ['kleinen', 'adj'], ['Kinder', 'nou']]
Morphological analysis:  [[['Mann', 'nou'], [['akk', 'mas'], ['dat', 'pl']]], [[
'machen', 'ver'], [['1', 'pl'], ['3', 'pl'], 'pres']], [['kleinen', 'adj'], [['n
om', 'pl'], ['akk', 'pl']]], [['Kinder', 'nou'], [['nom', 'pl'], ['akk', 'pl']]]
]
Disambiguated after noun-verb agreement:  [[['Mann', 'nou'], [['akk', 'mas'], ['
dat', 'pl']]], [['machen', 'ver'], [['3', 'pl'], 'pres']], [['kleinen', 'adj'],
[['nom', 'pl'], ['akk', 'pl']]], [['Kinder', 'nou'], [['nom', 'pl']]]]
Lemmatized:  [['Mann', ['Mann', 'Man']], ['machen', ['machen']], ['kleinen', ['k
lein']], ['Kinder', ['Kind']]]
Root translated:  [['den', 'the'], ['Mann', 'man'], ['machen', 'make'], ['die',
'the'], ['kleinen', 'small'], ['Kinder', 'child']]
NP Chunked English:  [[['the', 'art'], ['man', 'nou', [['akk', 'mas'], ['dat', '
pl']]]], ['make', 'ver', [['3', 'pl'], 'pres']], [['the', 'art'], ['small', 'adj
'], ['child', 'nou', [['nom', 'pl']]]]]
Inflected (only works before chunking):
['the', 'the'] ['man', ['akk', 'mas'], 'man'] ['man', ['dat', 'pl'], 'mans'] ['m
ake', ['3', 'pl'], 'make'] ['the', 'the'] ['small', 'small'] ['child', ['nom', '
pl'], 'childs']
Assigned an element type:
[[['the', 'art'], ['man', 'nou', [['akk', 'mas'], ['dat', 'pl']]], 'dobj'], ['ma
ke', 'ver', [['3', 'pl'], 'pres'], 'mverb'], [['the', 'art'], ['small', 'adj'],
['child', 'nou', [['nom', 'pl']]], 'sub']]
Assigned priority:
[['5', ['the', 'art'], ['man', 'nou', [['akk', 'mas'], ['dat', 'pl']]], 'dobj'],
 ['2', 'make', 'ver', [['3', 'pl'], 'pres'], 'mverb'], ['1', ['the', 'art'], ['s
mall', 'adj'], ['child', 'nou', [['nom', 'pl']]], 'sub']]
Rearranged to English structure:
[['1', ['the', 'art'], ['small', 'adj'], ['child', 'nou', [['nom', 'pl']]], 'sub
```

Figure 3: Running version of translation program.

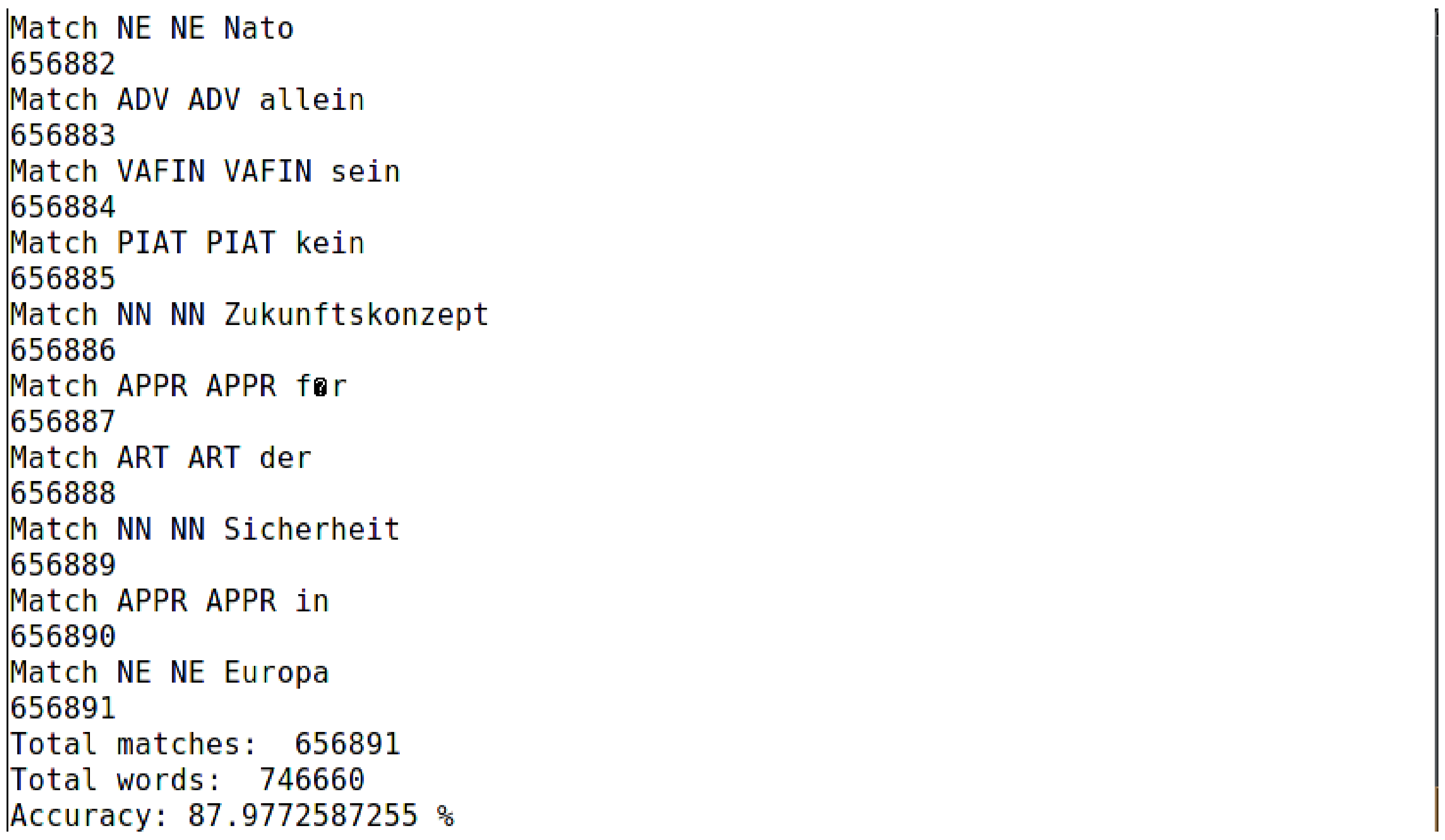
Grammar

In rule-based machine translation, parsing is the most difficult method to implement. In order to restructure simple German sentences to English ones, I assigned a priority number to each noun phrase chunk, based on where the chunk would appear in a n English sentence. The program then sorts based on priority number to restructure.



```
#BOS 22951 0 1071759059 0 %% (source: t_v_janilja 3158)
SPD-Spitze                  SPD-Spitze                  NN      Nom.Sg.F
00
stimmt                      stimmen                     VVFIN   3.Sg.Pre
00
Bosnien-Einsatz             Bosnien-Einsatz             NN      Dat.Sg.M
00
zu                          zu                          PTKVZ   --
00
#500                        --                          S       --
#EOS 22951
#BOS 22952 0 1071759080 0 %% (source: t_v_janbettina 678)
Parteitag                   Parteitag                   NN      Nom.Sg.M
04
soll                        sollen                      VMFIN   3.Sg.Pre
04
Engagement                  Engagement                  NN      Acc.Sg.N
02
deutscher                   deutsch                     ADJA    Pos.Gen.
00
Soldaten                    Soldat                      NN      Gen.Pl.M
00
``                          --                          $(      --
```

Figure 1: Excerpt from the TIGER Corpus, which contains over 700,000 entries.



```
Match NE NE Nato
656882
Match ADV ADV allein
656883
Match VAFIN VAFIN sein
656884
Match PIAT PIAT kein
656885
Match NN NN Zukunftskonzept
656886
Match APPR APPR f�r
656887
Match ART ART der
656888
Match NN NN Sicherheit
656889
Match APPR APPR in
656890
Match NE NE Europa
656891
Total matches:  656891
Total words:  746660
Accuracy: 87.9772587255 %
```

Figure 2: Running version of statistical methods.

Results

I ran my program on a series of input German sentences, and print out the results, with a correct translation for comparison of accuracy in translation and tagging. Statistical tagging verified the prediction that the program approaches 90% accuracy when each word is simply assigned its most frequently occurring tag. Rule-based methods only function correctly with grammatically correct simple sentences in "normal" sentence order, with words in regular positions – Subject, verb, object.