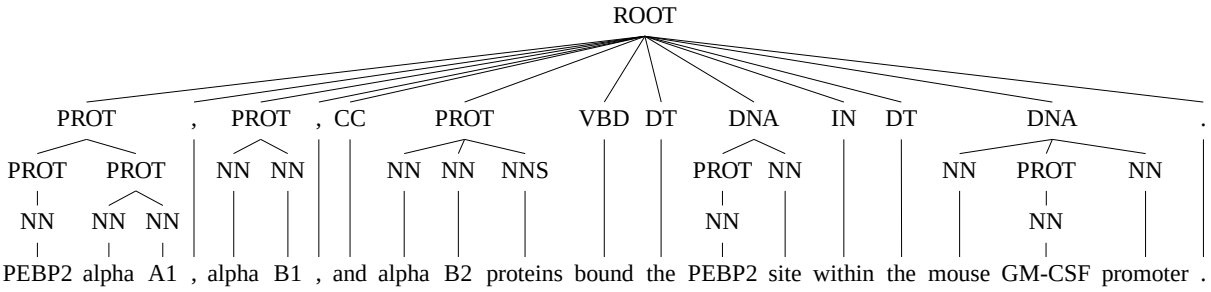

Figure 1: An example of our tree representation over nested named entities. The sentence is from the GENIA corpus. *PROT* is short for *PROTEIN*.

but they tend to be much flatter. This model allows us to include parts of speech in the tree, and therefore to jointly model the named entities and the part of speech tags. Once we have converted our sentences into parse trees, we train a discriminative constituency parser similar to that of (Finkel et al., 2008). We found that on top-level entities, our model does just as well as more conventional methods. When evaluating on *all* entities our model does well, with F-scores ranging from slightly worse than performance on top-level only, to substantially better than top-level only.

2 Related Work

There is a large body of work on named entity recognition, but very little of it addresses nested entities. Early work on the GENIA corpus (Kazama et al., 2002; Tsuruoka and Tsujii, 2003) only worked on the innermost entities. This was soon followed by several attempts at nested NER in GENIA (Shen et al., 2003; Zhang et al., 2004; Zhou et al., 2004) which built hidden Markov models over the innermost named entities, and then used a rule-based post-processing step to identify the named entities containing the innermost entities. Zhou (2006) used a more elaborate model for the innermost entities, but then used the same rule-based post-processing method on the output to identify non-innermost entities. Gu (2006) focused only on proteins and DNA, by building separate binary SVM classifiers for innermost and outermost entities for those two classes.

Several techniques for nested NER in GENIA where presented in (Alex et al., 2007). Their first approach was to layer CRFs, using the output of one as the input to the next. For inside-out layering, the first CRF would identify the innermost entities, the next layer would be over the words and the innermost entities to identify second-level entities, etc. For outside-in layering the first CRF would identify outermost entities, and then successive CRFs would identify increasingly nested entities. They also tried a cascaded approach, with separate CRFs for each entity type. The CRFs would be applied in a specified order, and then each CRF could utilize features derived from the output of previously applied CRFs. This technique has the problem that it cannot identify nested entities of the same type; this happens frequently in the data, such as the nested *proteins* at the beginning of the sentence in Figure 1. They also tried a joint labeling approach, where they trained a single CRF, but the label set was significantly expanded so that a single label would include all of the entities for a particular word. Their best results where from the cascaded approach.

Byrne (2007) took a different approach, on historical archive text. She modified the data by concatenating adjacent tokens (up to length six) into potential entities, and then labeled each concatenated string using the C&C tagger (Curran and Clark, 1999). When labeling a string, the "previous" string was the one-token-shorter string containing all but the last token of the current string. For single tokens the "previous" token was the longest concatenation starting one token earlier.

SemEval 2007 Task 9 (Márquez et al., 2007b) included a nested NER component, as well as noun sense disambiguation and semantic role labeling. However, the parts of speech and syntactic tree were given as part of the input, and named entities were specified as corresponding to noun phrases in the tree, or particular parts of speech. This restriction substantially changes the task. Two groups participated in the shared task, but only one (Márquez et al., 2007a) worked on the named entity component. They used a multi-label AdaBoost.MH algorithm, over phrases in the