Тестируемый парсер: TreeTagger

1. Анализ системы тегов
    a. Система учитывает 12 частей речи, но в ней нет причастий. Я бы их добавила, потому что, как показали тестирования, причастия классифицируются неоднозначно (то как глагол, то как прилагательное), хотя нельзя однозначно приписать их к одному или другому.
    b. Для местоимений есть отдельная часть речи в классификаторе.
    c. словоформы *нашедший* и *находившего*, *дал* и *давал* отнесены к разным частям речи:

нашедший	находившего	дал	давал
нашедший Vmps-smafpa найти	находившего Vmpp-smafeg находившего	Дал Vmis-sma-p дать	давал Vmis-sma-e давать

    d. Система тегов в TreeTagger достаточно простая, достаточно заменить условные обозначения TreeTagger на условные обозначения ЗС, а некоторые характеристики, учитываемые TreeTagger не брать, потому что их нет в ЗС.

2. Функциональное тестирование:
    a. Я создала файл со своими примерами сложных случаев, также, я смотрела и то, что получилось при разборе файла из архива.
    b. Проблемы токенизации:
        i. числа: классифицируются как числительные, иногда вместо леммы выдает “@card@”
        ii. десятичные и дробные числа: классифицируются как числительные, вместо леммы “@card@”
        iii. сокращения типа г.: если оно с пробелом, классифицируется как сокращение, если без, отдельно не классифицируется.
        iv. слова с дефисами: классифицируются как одно целое, но прилагательные через дефис не переводятся в лемму.
        v. слова с апострофом: классифицируются как одно целое, не род определяется как мужской.
        vi. знаки препинания: отделяются от слова, иногда классифицируются SENT
        vii. спецзнаки типа $ или &: так же как знаки препинания
        viii. вкраплениями другого алфавита или цифр: классифицируются в основном правильно. Но “к0т” и “1волк” почему-то отнесены к частице и прилагательному соответственно.
    c. Незнакомые слова:

        i. Обычно определяются незнакомые слова хорошо, если у них есть очевидные признаки принадлежности к какой-то части речи.
        ii. Если очевидных признаков нет, определяется как существительное или никак вообще
        iii. Лемма просто равна исходной словоформе.
        iv. NB - вообще для этого парсера леммы - проблема, потому что даже для знакомых ему слов он не всегда правильно ее определяет,а для сложных случаем, даже если классификация полностью получается, лемма = исходной словоформе
    d. Омонимичные словоформы:
        i. Предлагается только один вариант
        ii. не всегда правильный
        iii. Первая мысль: парсер выбирает наиболее частотный вариант из омонимичных разборов, НО слово “дорогой” было разобрано и как существительное и как прилагательное (причем оба раза не верно).
        iv. Я не нашла ни одной пары (вида) омонимичных словоформ, которые были бы разобраны хорошо.
3. Обработка файла:
    a. алгоритм работает однозначно
    b. лексическая точность - 0.868
    c. accuracy - 0.868