Проект по тестированию морфологических тагеров.

Подготовили: Атнагулова Залина, Зубова Дарья, Наумов Александр

1.1. Сколько частей речи учитывает система; какие части речи в системе отсутствуют, а Вы считаете, что эти части речи необходимо выделять (ответ мотивируйте)

Анализатор “распознает” 12 частей речи: существительные, глаголы, прилагательные, местоимения, наречия, предлоги, союзы, числительные, частицы, междометия/восклицания, аббревиатуры и “все остальное”, отдельный тег есть также у знаков конца предложения (.?!).

Нам кажется, что список достаточно исчерпывающий, но в примерах нами так и не был замечен тег “все остальное”, поэтому мы не очень поняли, для чего он используется. Возможно, туда можно было бы поместить смайлики и “запикивания” слов. Также нам кажется оправданным выделить отдельный тег для знаков препинания, потому что тагер различным запятым и двоеточиям не приписывает никаких показателей, а выдает что-то вроде “: -> - -> :” (стрелка вместо табуляции), или “, -> , -> ,”

1.2. В какие pos-классы попадают местоимения

Личные, возвратные, притяжательные местоимения разбираются и лемматизируются хорошо, нам кажется, что даже с очень высокой точностью.
Неопределенные, указательные, вопросительные, относительные, взаимные, определительные, отрицательные.
Взаимные местоимения все составные, поэтому разбираются не совсем корректно, так как токенизация в разметчике производится по словам. Например, “друг с другом” разбирается “друг” - местоимение, “с” - предлог, “другом” - местоимение.
Неопределенное местоимение “несколько” таггер отнес к числительным. В целом, с остальными неопределенными, указательными и отрицательными местоимениями у анализатора проблем не возникает, неизменяемые формы он помечает номинативом среднего рода, а форме “коими” в примере “коими они и являются” вполне оправданно присваивает творительный падеж.
С вопросительными и относительными местоименями много омонимии.
Относительное местоимение “что” довольно часто отделяется от омонимичного союза “что”, это хорошо, но при этом, чтобы определить “что” как местоимение, как нам показалось, анализатору требуется идущее рядом сказуемое. Так, во фразе типа “определите, ***что*** происходит с числами” он правильно определит тег, а в выражении “существенно, что второй глаз при этом не закрывается, ***что*** позволяет лучше

контролировать обстановку” относительное местоимение анализируется таггером как союз, хоть рядом и есть сказуемое (может это вызвано наличием союза “что” неподалеку).

1.3. Как лемматизируются причастия?

Как и ожидалось, с лемматизацией причастий возникает очень много проблем. С одной стороны, для большинства кратких причастий таггер срабатывает очень классно, например:

Слово	**Разбор**	**Лемма**
отнесены	Vmps-p-psp	отнести
разработана	Vmps-sfpsp	разработать
предназначена	Vmps-sfpsp	предназначить

Для полных же причастий существуют некоторые особенности. Например, часто причастия определяются как отглагольные прилагательные: у нас был пример “на ***согнутых*** в локтях руках”, получился разбор: согнутых Afpmplf согнутый. Однако если поставить причастие после определяемого слова, то есть “рука, ***согнутая*** в локте”, то разбор будет таким: согнутая Vmps-spf-pn согнуть. Но такое часто только с причастиями, больше похожими на отглагольные прилагательные. В случаях же когда причастие “явное”, таггер определяет его достаточно хорошо. НО! Очень часто он предъявляет неправильную лемму, нередко просто оставляет поданную ему словоформу без изменений:

Слово	**Разбор**	**Лемма**
проспавших	Vmps-p-afpg	проспавших
усложняющие	Vmpp-p-afea	усложняющие
находившего	Vmpp-smafeg	находившего

Но при этом стоит заметить, что все же как минимум в половине случаев с причастиями таггер со своей работой справляется и дает верный тег и верную лемму.

1.4. К одной или разным леммам будет отнесены словоформы *нашедший* и *находившего*, *дал* и *давал*

Слово	**Разбор**	**Лемма**

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

1.5. Напишите правило пересчета тегов системы на теги из ЗС для анафорических местоимений (*он, она* и т.п.) и наречий

Для решения этой задачи, был написан скрипт **“пункт 1.5.py”**, который меняет формат записи разбора.

2.2.1. Как решаются проблемы токенизации: что происходит с числами, десятичными числами, сокращениями типа г., словами с дефисами, апострофом, знаками препинания? спецзнаками типа $ или &, смешанными элементами (буквы+цифры, вкраплениями другого алфавита) etc.?

Токенизация по пробелам, все знаки препинания, тире, он старается выделять отдельно, точка, вопросительный и восклицательный знаки помечаются как границы предложений, все остальные не помечаются никак.

У чисел анализатор отделяет точку от числа только если их больше одной, в нумерованном списке 1. и 1.1 останутся с точкой, причем второе он, видимо, посчитает числом с десятичной частью, а первое разберет как Sent, что означает “граница предложения”, а вот у 1.1. последняя точка будет вынесена в отдельную словоформу, так же Sent.

Все сокращения он так же разбивает по точкам и оставшуюся часть пытается восстановить, так, “г.” он всегда расшифровывает как “год”, а “т.п.” как “то + подобный”, если он никак не может восстановить слово сокращения, он оставит его таким, какое оно в тексте, и может разобрать его как предлог, например в случае “у.е.” у для него просто предлог, а вот про е он думает, что это сокращение от глаголы “быть”.

Если подать ему для разбора просто набор символов, например “Уметсн&&оя**”, если в нем есть буквы, он попытается разобрать его по каким-то формальным признакам, по окончанию, приставке, корню, окружению, если такое есть, и все-таки разобрать, наш пример - это полное прилагательное мужского рода в номинативе единственного числа. Вообще, он старается не отделять затисавшиеся внутри слова посторонние знаки и как-то разбираться прямо с ними. Но ссылки он распознает и выделяет отдельным токеном “URLTOKEN”, правда может разбить его на два токена, например “U” и “RLTOKEN”.

Если в слово вкрадывается одна буква латиницы, похожая на ту букву кириллицы, которая должна быть в слове, анализатор сможет сможет распознать слово правильно, но в примере вроде “bcegda” он скажет, что это числительное.

2.2.2. Что происходит с незнакомыми словами? Насколько хорошо предсказываются их грамматические характеристики, их леммы?

С незнакомыми словами он разбирается не очень хорошо. Мы проверили анализатор на “глокой куздре” и первом рассказе о “Пуськах бятых” и обнаружили, что одушевленность и неодушевленность приписывается совершенно случайным образом, что таггер ошибается в единичных незнакомых словах, но когда они идут цельным текстом, ему довольно хорошо удается “прочувствовать” его язык, если он мотивирован, и он практически не ошибается в разметке. Но вот леммы таггер совсем не пытается восстанавливать по получившимся разборам, даже считая какое-то слово прилагательным в женском роде в номинативе, он не попытается образовать от него форму мужского рода, чтобы обозначить её в качестве леммы, что на первый взгляд, казалось бы, стоило сделать.

2.2.3. Что происходит с омонимичными словоформами: предлагается только один максимально вероятный вариант, предлагаются все возможные варианты, предлагаются все варианты, за исключением очень маловероятных случаев или случаев, снимаемых "надежными" правилами и т.п.

Таггер предлагает только один вариант, скорее всего, максимально вероятный. Но, к сожалению, он не всегда оказывается правильным, как, например, уже было сказано выше про словоформу “что”.

2.2.4. Какие проблемные случаи омонимичных разборов разбираются хорошо, в каких часто возникают ошибки и т.п. (например, (а) частеречная омонимия: прилагательное vs. существительное, глагол vs. прилагательное, наречие vs. частица; (б) падежная омонимия; (в) омонимия различных местоименных форм и т.д.)

В целом нам кажется, что для анализатора с омонимией он справляется довольно хорошо. Однако можно заметить тенденцию, что в проблемных случаях, особенно с несуществующими словами он любит называть слово прилагательным, если у него, скажем, окончание, хоть немного напоминающее окончание прилагательного.

Конечно же тагер ошибается в распознавании омонимичных по форме падежей, и ему в этом, к сожалению, не помогает даже наличие стоящего рядом имени, форма которого в данном падеже не омонимична. Например в предложении “Красивой розе дали страшное имя” он припишет прилагательному творительный падеж, а “розе” - дательный.

Насколько мы поняли, различные виды местоимений он не определяет совсем, хотя для них есть теги, в остальном же он довольно хорошо ловит падеж, род и лицо местоимения.

Часть 3.

1.3.

С помощью морфологического анализатора был обработан файл textBLOG.txt

Результаты обработки оказались довольно позитивными.

Из 500 словоформ:

- 12 Лемм неправильно подобраны
- 32 Неправильный разбор(тег)
- 66 Нет разбора вообще

Уровень оставшейся неоднозначности: число элементов в Output(W) для всех слов тестируемого текста, поделенное на число слов в тексте. Если алгоритм работает однозначно, то этот параметр равняется 1.

Слов в тексте 500.

У каждого слова есть 2 или 1 элементов (разбор, лемма). В нашем случае 2 элемента есть у 434 слов. Т.е. элементов 434 * 2 = 868.

868 /500 = 1,736

Не уверены, что правильно посчитано. Не до конца понятна формулировка вопроса.

Лексическая точность алгоритма = 0,972

Точность = 0,923

Accuracy = 0,936