Аннотация

В данной работе исследована возможность использования семантико-синтаксического анализатора Compreno в качестве источника высокоуровневых признаков для решения задачи распознавания именованных сущностей в рамках нейросетевого подхода. Исследование проводилось на англоязычном корпусе CoNLL 2003. Полученные результаты показывают, что высокоуровневые признаки дают ощутимый прирост оценки качества, без какой-либо инженерии над ними.

Ключевые слова: нейронные сети, распознавание именованных сущностей.

Содержание

Глоссарий

Распределённый — Слово, которое нельзя употреблять. Но надо протестировать длинные строки в глоссарии.

Обозначения и сокращения

АИС — Автоматизированная информационная система. Но надо протестировать длинные строки в определениях.

Введение

Именованная сущность - это слово или словосочетание обозначающее предмет или явления определенной категории. Примерами именованных сущностей являются имена людей, названия организаций и локаций. Задача распознавания именованных сущностей (Named Entity Recognition, NER) состоит в выделении и классификация именованных сущностей в тексте. В рамках конференции CoNLL 2003 проводилось соревнование для оценки качества методов распознавания именованных сущностей четырех типов на англоязычном корпусе [Tjong Kim Sang and De Meulder 2003]. Для решения задачи NER предлагалось много разных подходов [Nadeau and Sekine 2007]. В последнее время было показано, что методы на основе нейронных сетей показывают лучшие результаты для различных языков и корпусов, включая CoNLL 2003 [Yang et al. 2016].

Вместо большого количества вручную построенных признаков решающих определенную задачу, нейросетевые методы используют универсальные векторные представления слов [Mikolov et al. 2013]. Согласно гипотезе о дистрибутивности, эти представления кодируют в себе смысл слов [Sahlgren 2008]. Это позволяет строить мультизадачные и языконезависимые архитектуры [Collobert et al. 2011, Yang et al. 2016].

Несмотря на то, что использование универсальных векторных представлений получило в последнее время огромную популярность в силу своей эффективности и огромной экономии человеческих усилий, большой интерес все еще представляет исследование возможностей использования высокоуровневых признаков в качестве входных данных для нейросетей. Так, например, в работах [Xu et al. 2014, Bian et al. 2014] описано использование морфологических, синтаксических и семантических признаков для построения более совершенных векторных представлений слов.

Compreno - это технология автоматического анализа текстов на естественном языке, в основе которой лежит многоуровневое лингвистическое описание, создававшееся профессиональными лингвистами в течение длительного времени [Anisimovich et al. 2012]. Помимо ручного описания Compreno использует для анализа большое количество инфор-

мации, извлекаемых различными статистическими методами из текстовых корпусов. В Compreno реализована процедура семантико-синтаксического анализа текста, в результате которой любому предложению на естественном языке (английском или русском) ставится в соответствие семантико-синтаксическое дерево, моделирующее смысл предложения и содержащее грамматическую и семантическую информацию о каждом слове предложения.

В данной работе исследована возможность использования семантико-синтаксического анализатора Compreno в качестве источника высокоуровневых признаков для задачи NER на корпусе CoNLL 2003 в рамках нейросетевого подхода.

Статья организована следующим образом: в части 1 проведен обзор связанных работ. Выбранная нейросетевая модель и способы внедрения синтактико-семантических признаков описаны в части 2. В части 3 описаны проведенные эксперименты и программная реализация.

Полученные результаты показывают повышение F1-меры почти на 1% на корпусе CoNLL 2003 при использовании синтактико-семантических признаков Compreno (87.49% против 88.47%). При этом затраты на их внедрение были минимальными - инженерия над признаками не проводилась.

1 Аналитический раздел

В данном разделе анализируется и классифицируется существующая всячина и пути создания новой всячины. А вот отступ справа в 1 см. — это хоть и по ГОСТ, но ведь диагноз же...

1.1 Анализ того и сего

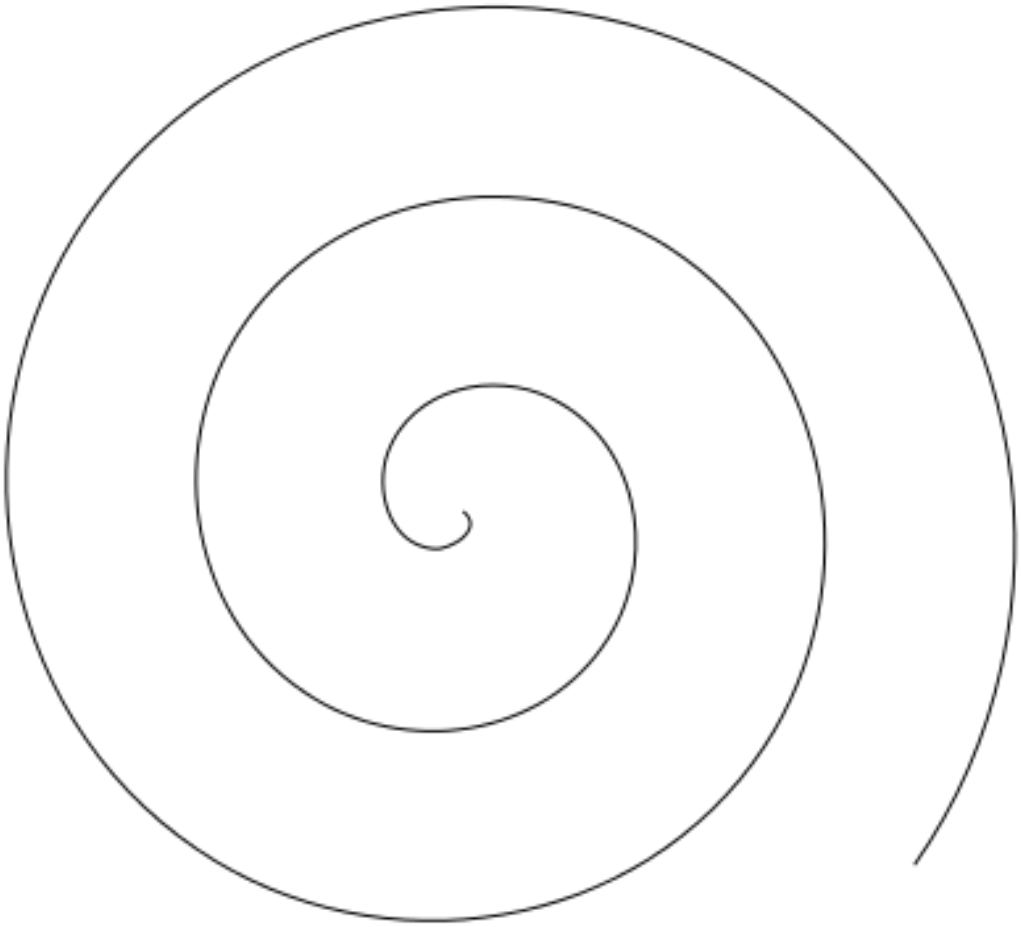

Рисунок 1.1 — Рисунок

В **?** указано, что...

Кстати, про картинки. Во-первых, для фигур следует использовать `[ht]`. Если и после этого картинки вставляются «не по ГОСТ», т.е. слишком далеко от места ссылки, — значит у вас в РПЗ **слишком мало текста**! Хотя и ужасный параметр `!ht` у окружения `figure` тоже

никто не отменял, только при его использовании документ получается страшный, как в ворде, поэтому просьба так не делать по возможности.

1.2 Существующие подходы к созданию всячины

Известны следующие подходы...

а) Перечисление с номерами.

б) Номера первого уровня. Да, ГОСТ требует именно так — сначала буквы, на втором уровне — цифры. Чуть ниже будет вариант «нормальной» нумерации и советы по её изменению. Да, мне так нравится: на первом уровне выравнивание элементов как у обычных абзацев. Проверим теперь вложенные списки.

1) Номера второго уровня.
2) Номера второго уровня. Проверяем на длиииином-предлиииииииииинной строке, что получается.... Сойдёт.

в) По мнению Лукьяненко, человеческий мозг старается подвести любую проблему к выбору из трех вариантов.

г) Четвёртый (и последний) элемент списка.

Теперь мы покажем, как изменить нумерацию на «нормальную», если вам этого захочется. Пара команд в начале документа поможет нам.

1) Изменим нумерацию на более привычную...
2) ... нарушим этим гост.
   а) Но, пожалуй, так лучше.

В заключение покажем произвольные маркеры в списках. Для них нужен пакет **enumerate**.

1. Маркер с арабской цифрой и с точкой.
2. Маркер с арабской цифрой и с точкой.
   I. Римская цифра с точкой.
   II. Римская цифра с точкой.

В отчётах могут быть и таблицы — см. табл. 1.1 и 1.2. Небольшая таблица делается при помощи **tabular** внутри **table** (последний полностью аналогичен **figure**, но добавляет другую подпись).

Таблица 1.1 — Пример короткой таблицы с длинным названием на много длинных-длинных строк

Тело	F	V	E	$F+V-E-2$
Тетраэдр	4	4	6	0
Куб	6	8	12	0
Октаэдр	8	6	12	0
Додекаэдр	20	12	30	0
Икосаэдр	12	20	30	0
Эйлер	666	9000	42	$+\infty$

Для больших таблиц следует использовать пакет **longtable**, позволяющий создавать таблицы на несколько страниц по ГОСТ.

Для того, чтобы длинный текст разбивался на много строк в пределах одной ячейки, надо в качестве ее формата задавать `p` и указывать явно ширину: в мм/дюймах (`110mm`), относительно ширины страницы (`0.22\textwidth`) и т.п.

Можно также использовать уменьшенный шрифт — но, пожалуйста, тогда уж во **всей** таблице сразу.

Таблица 1.2 — Пример длинной таблицы с длинным названием на много длинных-длинных строк

Вид шума	Громкость, дБ	Комментарий
Порог слышимости	0	
Шепот в тихой библиотеке	30	
Обычный разговор	60-70	
Звонок телефона	80	Конечно, это было до эпохи мобильников
Уличный шум	85	(внутри машины)
Гудок поезда	90	

Продолжение на след. стр.

Продолжение таблицы 1.2

Шум электрички	95	
Порог здоровой нормы	90-95	Длительное пребывание на более громком шуме может привести к ухудшению слуха
Мотоцикл Power Mower Бензопила Рок-концерт	100 107 110 115	 (модель бензокосилки) (Doom в целом вреден для здоровья)
Порог боли	125	feel the pain
Клепальный молоток	125	(автор сам не знает, что это)
Порог опасности	140	Даже кратковременное пребывание на шуме большего уровня может привести к необратимым последствиям
Реактивный двигатель Самый громкий возможный звук	140 180 194	 Необратимое полное повреждение слуховых органов Интересно, почему?..

2 Конструкторский раздел

В данном разделе проектируется новая всячина.

2.1 Архитектура всячины

Проверка параграфа. Вроде работает.

Вторая проверка параграфа. Опять работает.
Вот.

— Это список с «палочками».
— Хотя он и не по ГОСТ, кажется.

1) Поэтому для списка, начинающегося с заглавной буквы, лучше список с цифрами.

Формула 2.1 совершено бессмысленна.

$$a = cb \tag{2.1}$$

Окружение `cases` опять работает (см. 2.2), спасибо И. Короткову за исправления..

$$a = \begin{cases} 3x + 5y + z, \text{если хорошо} \\ 7x - 2y + 4z, \text{если плохо} \\ -6x + 3y + 2z, \text{если совсем плохо} \end{cases} \tag{2.2}$$

2.2 Подсистема всякой ерунды

Культурная вставка dot-файлов через утилиту dot2tex (рис. 2.1).

2.2.1 Блок-схема всякой ерунды

Кстати о заголовках

У нас есть и **subsubsection**. Только лучше её не нумеровать.

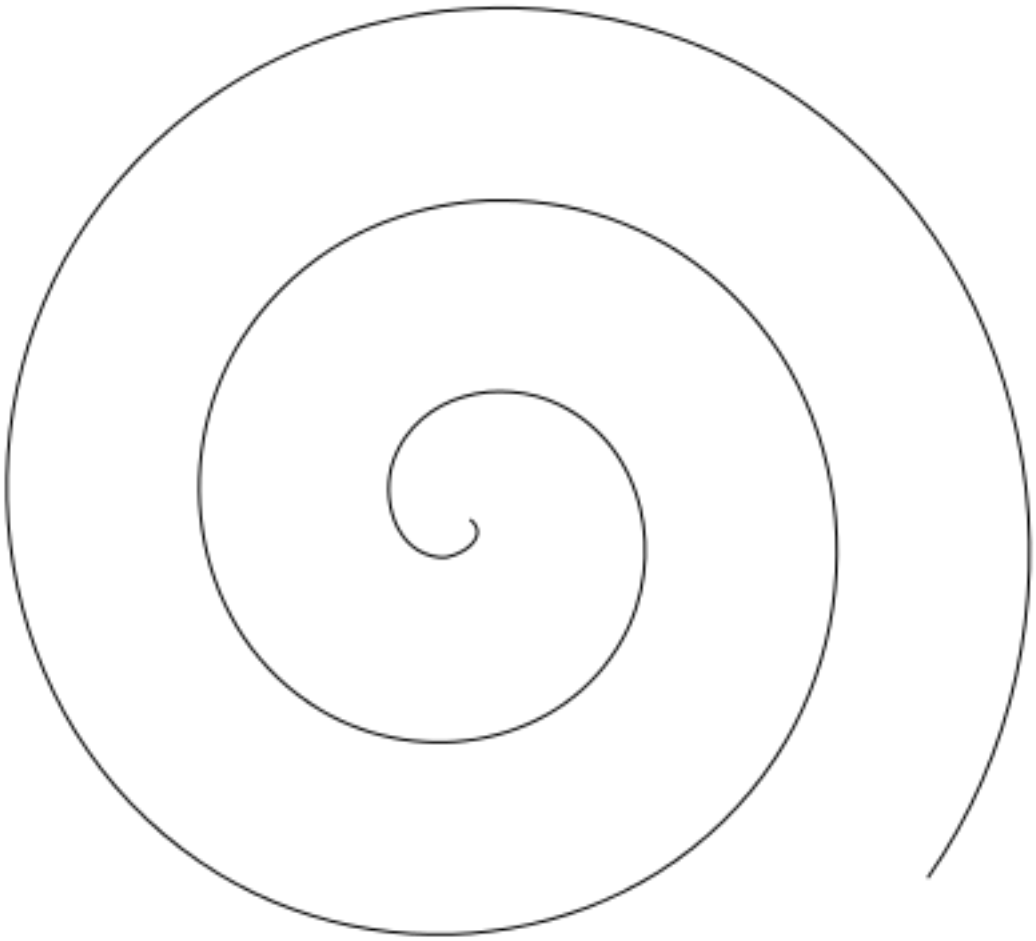

Рисунок 2.1 — Рисунок

3 Технологический раздел

В данном разделе описано изготовление и требование всячины. Кстати, в Latex нужно эскейпить подчёркивание (писать «`some\_function`» для **some_function**).

Для вставки кода есть пакет **listings**. К сожалению, пакет **listings** всё ещё работает криво при появлении в листинге русских букв и кодировке исходников utf-8. В данном примере он (увы) на лету конвертируется в koi-8 в ходе сборки pdf.

Есть альтернатива **listingsutf8**, однако она работает лишь с **lstinputlisting**, но не с окружением **lstlisting**

Вот так можно вставлять псевдокод (питоноподобный язык определен в **listings.inc.tex**):

Листинг 3.1 — Алгоритм оценки дипломных работ

```
def EvaluateDiplomas():
    for each student in Masters:
        student.Mark ← 5
    for each student in Engineers:
        if Good(student):
            student.Mark ← 5
        else:
            student.Mark ← 4
```

Еще в шаблоне определен псевдоязык для BNF:

Листинг 3.2 — Грамматика

```
  ifstmt → "if" "(" expression ")" stmt  |
           "if" "(" expression ")" stmt1 "else" stmt2
  number → digit digit *
```

В листинге 3.3 работают русские буквы. Сильная магия. Однако, работает только во включаемых файлах, прямо в TEX нельзя.

Листинг 3.3 — Пример (**test.c**)

```
#include <stdio.h>
int main()
{
    return 0;
}
```

Можно также использовать окружение **verbatim**, если **listings** чем-то не устраивает. Только следует помнить, что табы в нём «съедаются». Существует так же команда **verbatiminput** для вставки файла.

```
a_b = a + b; // русский комментарий
if (a_b > 0)
    a_b = 0;
```

4 Экспериментальный раздел

В данном разделе проводятся вычислительные эксперименты. А на рис. 4.1 показана схема мыслительного процесса автора...

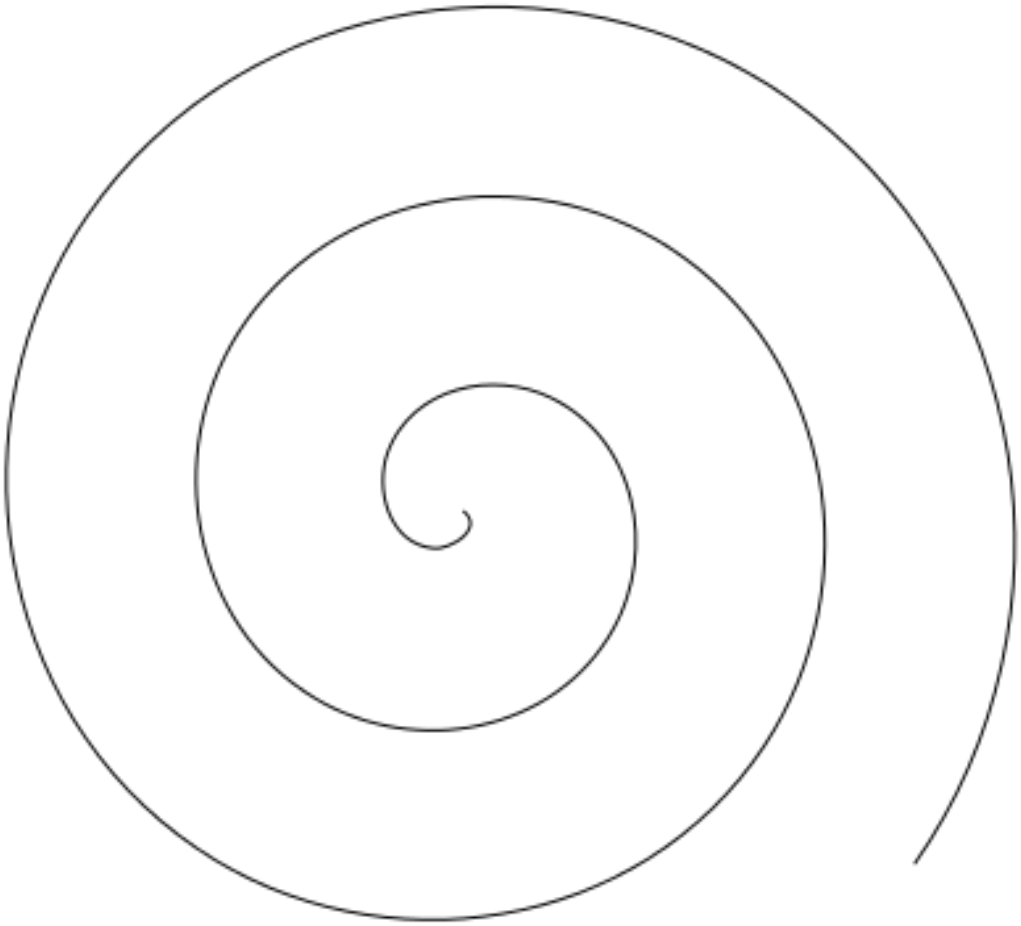

Рисунок 4.1 — Как страшно жить

Заключение

В результате проделанной работы стало ясно, что ничего не ясно...

Список использованных источников

Anisimovich, K. V., Druzhkin, K. J., Minlos, F. R., Petrova, M. A., Selegey, V. P., and Zuev, K. A. (2012). Syntactic and semantic parser based on ABBYY Compreno linguistic technologies. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference Dialog*, page 18.

Bian, J., Gao, B., and Liu, T.-Y. (2014). Knowledge-powered deep learning for word embedding. In *Machine Learning and Knowledge Discovery in Databases*, pages 132–148. Springer.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Sahlgren, M. (2008). The distributional hypothesis. from context to meaning: Distributional models of the lexicon in linguistics and cognitive science (special issue of the italian journal of linguistics). *Rivista di Linguistica*, 20(1).

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Xu, C., Bai, Y., Bian, J., Gao, B., Wang, G., Liu, X., and Liu, T.-Y. (2014). Rc-net: A general framework for incorporating knowledge into

word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1219–1228. ACM.

Yang, Z., Salakhutdinov, R., and Cohen, W. (2016). Multi-task cross-lingual sequence tagging from scratch. *CoRR*, abs/1603.06270.

Приложение А Картинки

Рисунок А.1 — Картинка в приложении. Страшная и ужасная.

Приложение Б Еще картинки

Рисунок Б.1 — Еще одна картинка, ничем не лучше предыдущей. Но надо же как-то заполнить место.