Анализ неструктурированных данных

Семинар 3

SENNA, Томита-парсер, отношения между словами

Национальный Исследовательский Университет
Высшая Школа Экономики

18-24 сентября 2017

План на сегодня

1. SENNA
2. Скрытые марковские модели
3. CMM для SENNA
4. Томита-парсер
5. Отношения между словами

План на сегодня

1. SENNA
2. Скрытые марковские модели
3. CMM для SENNA
4. Томита-парсер
5. Отношения между словами

SENNA (Semantic Extraction using a Neural Network Architecture) – система (а также архитектура нейронной сети), применяющаяся для различных задач анализа текста, в частности, SRL (Semantic Role Labeling).

- Базируется на принципах сверточных нейронных сетей.
- Разработана и представлена в 2007-2008, однако до сих пор является одной из самых удобных и совершенных систем.

SENNA

Из документации:

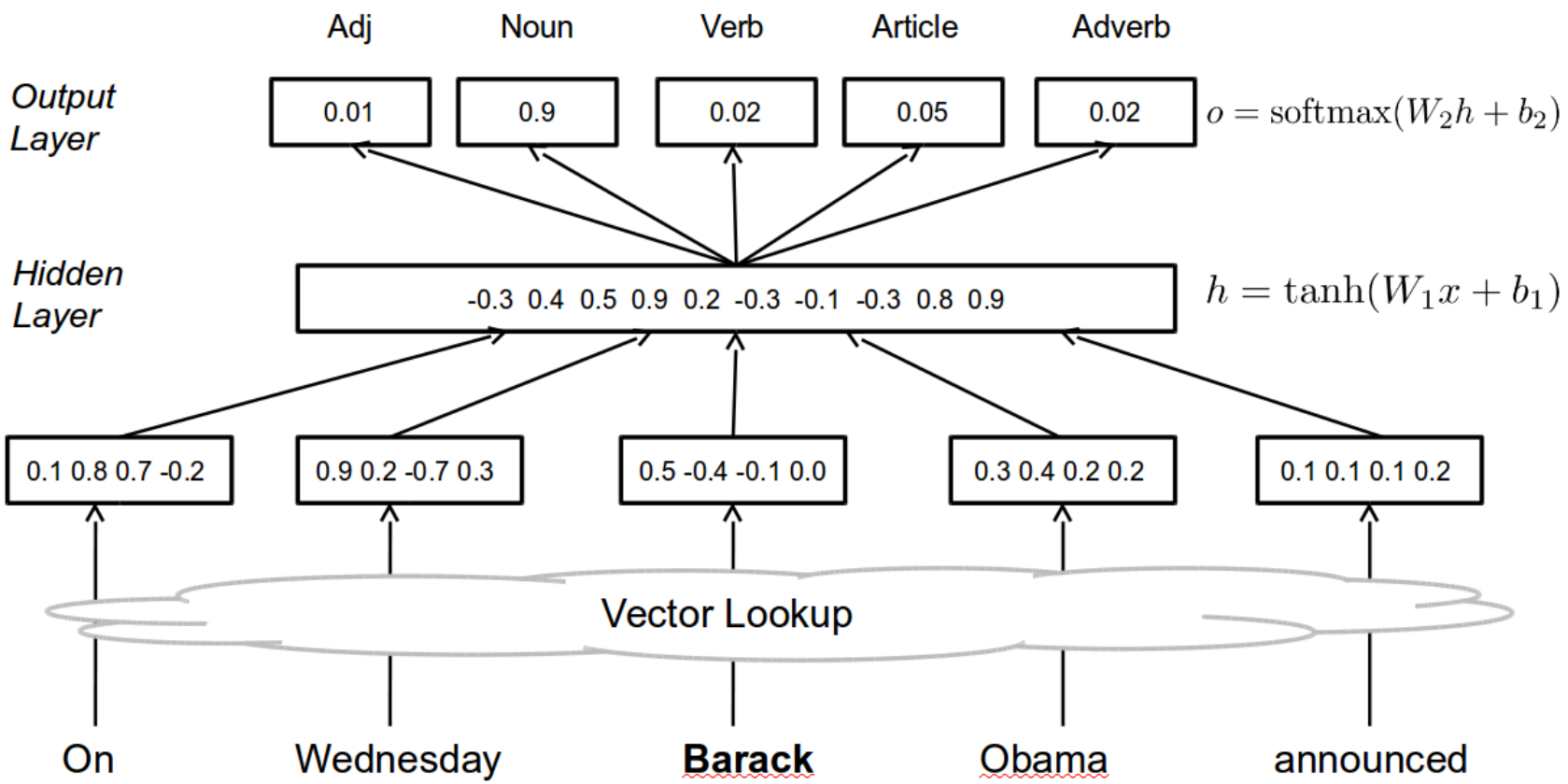
- SENNA is a software distributed under a non-commercial license, which outputs a host of Natural Language Processing (NLP) predictions: part-of-speech (POS) tags, chunking (CHK), name entity recognition (NER), semantic role labeling (SRL) and syntactic parsing (PSG).
- SENNA is fast because it uses a simple architecture, self-contained because it does not rely on the output of existing NLP system, and accurate because it offers state-of-the-art or near state-of-the-art performance.
- SENNA is written in ANSI C, with about 3500 lines of code. It requires about 200MB of RAM and should run on any IEEE floating point computer.

https://ronan.collobert.com/senna/

Система применяется для различных задач классификации текстов на уровне слов и предложений (POS-tagging, NER, Chunking). Два способа применения:

- Window-Approach: необходимая информация содержится в контексте слов (NER, POS)
- Sentence-Approach: важно рассматривать предложения целиком (разбор предложений)

SENNA - схема


Adj
Noun
Verb
Article
Adverb
Output Layer
0.01
0.9
0.02
0.05
0.02
$o = \text{softmax}(W_2 h + b_2)$
Hidden Layer
-0.3 0.4 0.5 0.9 0.2 -0.3 -0.1 -0.3 0.8 0.9
$h = \tanh(W_1 x + b_1)$
0.1 0.8 0.7 -0.2
0.9 0.2 -0.7 0.3
0.5 -0.4 -0.1 0.0
0.3 0.4 0.2 0.2
0.1 0.1 0.1 0.2
Vector Lookup
On
Wednesday
Barack
Obama
announced

SENNA - Vector Lookup

Этап получения векторного представления слов (word embedding). На этом этапе могут быть следующие действия:

- Преобразование из формата BoW.
- Преобразования из различных word embedding к формату, требуемому входом сети.

SENNA - слои сети

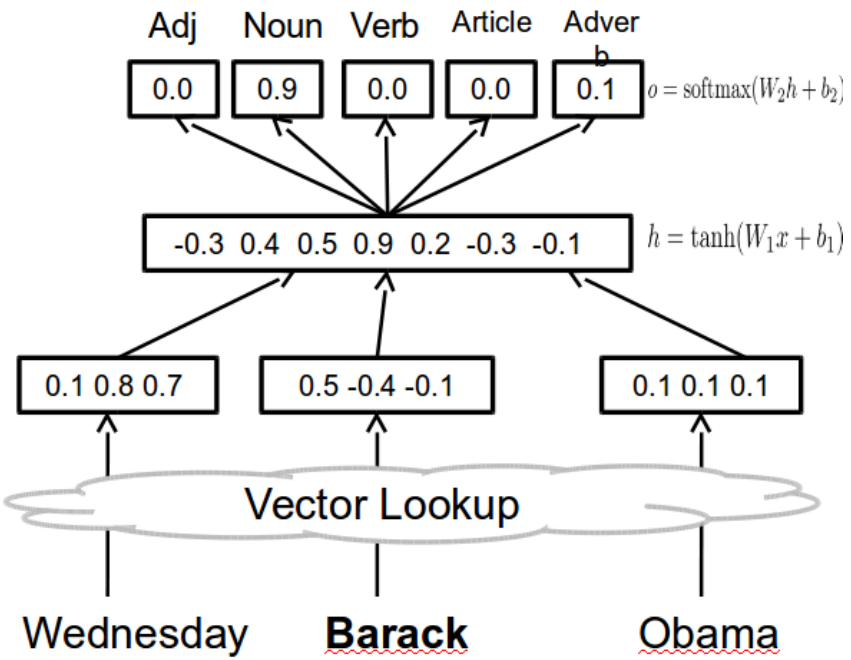
В сети присутствуют как минимум два слоя:

- Слой для поиска локальных фич среди соседних слов, использует сконкатенированные вектора слов
- Слой для поиска глобальных оптимумов на уровне предложения – дает ответ к основной задаче

Подробное описание архитектуры:
https://ronan.collobert.com/pub/matos/2009_tutorial_nips.pdf,
http://ml.nec-labs.com/software.php?project=senna.

SENNA - пример

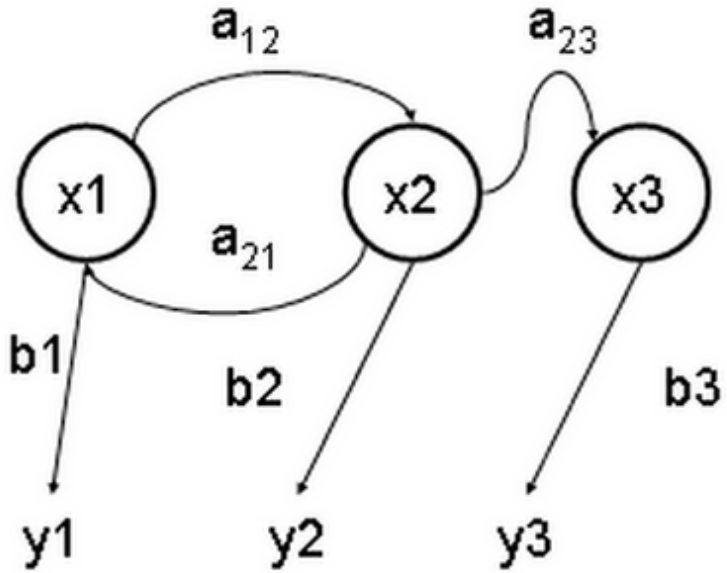
1. Select a target word, e.g. **Barack**
2. Create a window of *n* tokens around the target. A window-size=1 would create the tuple (Wednesday, **Barack**, Obama)
   - Use special PADDING token for tokens at sentence border, e.g. (PADDING, **Wednesday**, Barack)
3. Map each token to its word embedding (e.g. 100 dim. word embedding)
4. Concatenate the 3x100 dim. vectors and feed the 300 dim. vector into a dense hidden layer and apply tanh-activation function
5. Take the output of the hidden layer, feed it into a dense layer and apply the softmax-activiation function



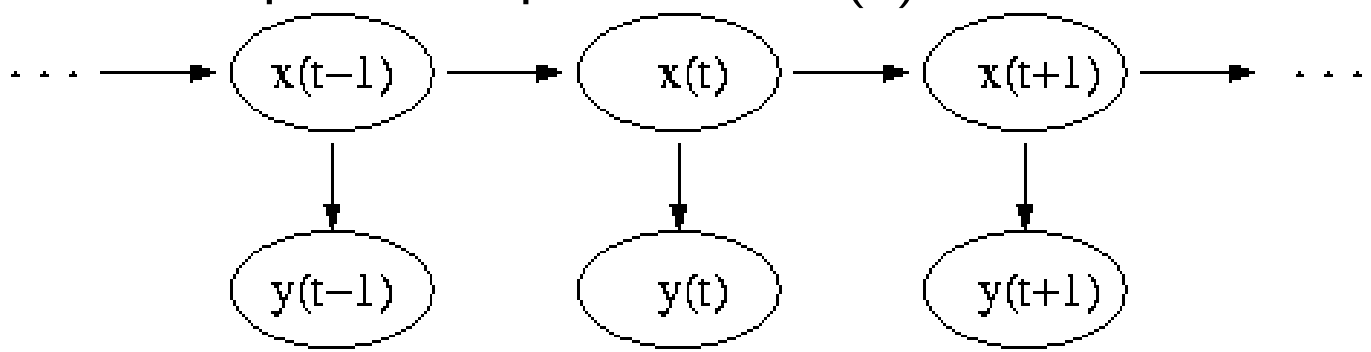
План на сегодня

1. SENNA
2. Скрытые марковские модели
3. CMM для SENNA
4. Томита-парсер
5. Отношения между словами

Скрытая марковская модель (СММ) – это вероятностная модель последовательности, имитирующая работу процесса, похожего на марковский процесс с неизвестными параметрами. Задачей ставится разгадывание неизвестных параметров на основе наблюдаемых



В СММ можно наблюдать переменные, на которые оказывают влияние состояния модели. Случайная переменная $x(t)$ – значение скрытой переменной в момент времени t, а $y(t)$ – значение наблюдаемой переменной в момент времени t. Значение скрытой переменной $x(t)$ зависит только от значения скрытой переменной $x(t-1)$, а значение наблюдаемой переменной $y(t)$ зависит только от значения скрытой переменной $x(t)$.



Алгоритмы решения задач, связанных с CMM:

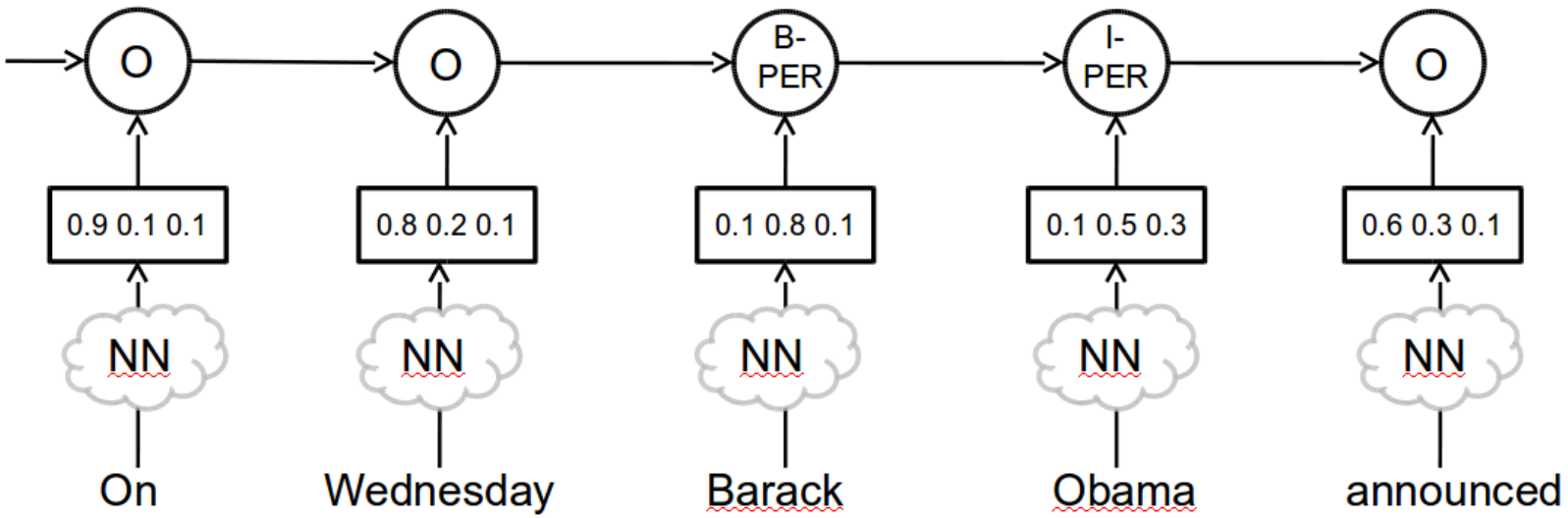
- Алгоритм прямого-обратного хода: даны параметры модели и последовательность, требуется вычислить вероятность появления данной последовательности.
- Алгоритм Витерби: даны параметры модели, требуется определить наиболее подходящую последовательность скрытых узлов, наиболее точно описывающую данную модель.
- Алгоритм Баума-Велша: дана выходная последовательность (или несколько) с дискретными значениями, требуется «потренировать» CMM на данном выходе.

План на сегодня

1. SENNA
2. Скрытые марковские модели
3. CMM для SENNA
4. Томита-парсер
5. Отношения между словами

Предложена модификация Sentence-Tag-Criterion (STC, Collobert et al.), добавляющая скрытую марковскую модель над нейронной сетью. Подход оптимизирует тэггинг предложений, демонстрирует отличные результаты. На данный момент нет реализации метода «из коробки».

SENNA - HMM


O
O
B-
PER
I-
PER
O
0.9 0.1 0.1
0.8 0.2 0.1
0.1 0.8 0.1
0.1 0.5 0.3
0.6 0.3 0.1
NN
NN
NN
NN
NN
On
Wednesday
Barack
Obama
announced

Практическое применение SENNA:

- POS-tagger, использующий SENNA: https://github.com/biplab-iitb/practNLPTools.
- Large Scale Application of Neural Network Based Semantic Role Labeling for Automated Relation Extraction from Biomedical Texts (http://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0006393type=printable)
- ...

План на сегодня

1. SENNA
2. Скрытые марковские модели
3. CMM для SENNA
4. Томита-парсер
5. Отношения между словами

Томита-парсер

Система, предназначенная для извлечения структурированных сущностей из текста: фактов, отношений, причинно-следственных связей и др.

- Использует механизм контекстно-свободных грамматик и словари ключевых слов.
- Позволяет добавлять свои расширения.

https://tech.yandex.ru/tomita/, исходный код: https://github.com/yandex/tomita-parser/

Пример:

$$S- > Noun$$

```
t = 'механизм контекстно-свободных грамматик'
...
['механизм', 'грамматика']
```

Описание простейшей грамматики:

```
#encoding "utf-8"
#GRAMMAR_ROOT S
S -> Noun;
```

Сохраняется в специальный файл (first.cxx)

Файл mydic.gzt – корневой словарь:

```
encoding "utf8";
import "base.proto";              // описания protobuf-типов (TAuxDicA
rticle и прочих)
import "articles_base.proto";  // Файлы base.proto и articles_base.
proto встроены в компилятор.
                                  // Их необходимо включать в начало л
юбого gzt-словаря.
// статья с нашей грамматикой:
TAuxDicArticle "наша_первая_грамматика"
{
    key = { "tomita:first.cxx" type=CUSTOM }
}
```

Файл config.proto:

```
encoding "utf8";
TTextMinerConfig {
  Dictionary = "mydic.gzt"; // путь к корневому словарю
  PrettyOutput = "PrettyOutput.html"; // путь к файлу с отладочным
 выводом в удобном для чтения виде
  Input = {
    File = "test.txt"; // путь к входному файлу
  }
  Articles = [
    { Name = "наша_первая_грамматика" } // название статьи в корнев
ом словаре,
                                        // которая содержит запус
каемую грамматику
  ]
}
```

Осталось сделать файл test.txt и проверить:

запуск:

```
./tomitaparser config.proto
```

Вывод печатается в файл PrettyOutput.html.

Труд облагораживает человека . **EOS**

Text	Type
труд	TAuxDicArticle [наша_первая_грамматика]
человек	TAuxDicArticle [наша_первая_грамматика]

Составные правила в грамматике и терминальные токены:

$$S -> Adj\ \ Noun;$$

$$S -> Adj\ \ "word";$$

Подробная документация:
https://tech.yandex.ru/tomita/doc/tutorial/concept/basic-rules-docpage/

План на сегодня

1. SENNA
2. Скрытые марковские модели
3. CMM для SENNA
4. Томита-парсер
5. Отношения между словами

Типы отношений

Главные типы отношений:

- Синонимы – это слова, разные по написанию, но имеющие схожее или тождественное значение: смелый – храбрый.
- Гипоним – понятие, выражающее частную сущность по отношению к другому, более общему понятию: собака – мопс.
- Гипероним – понятие в отношении к другому понятию, выражающее более общую сущность. В отношении некоторого множества объектов гиперонимом является понятие, отражающее надмножество к исходному: собака – млекопитающее.

Типы отношений

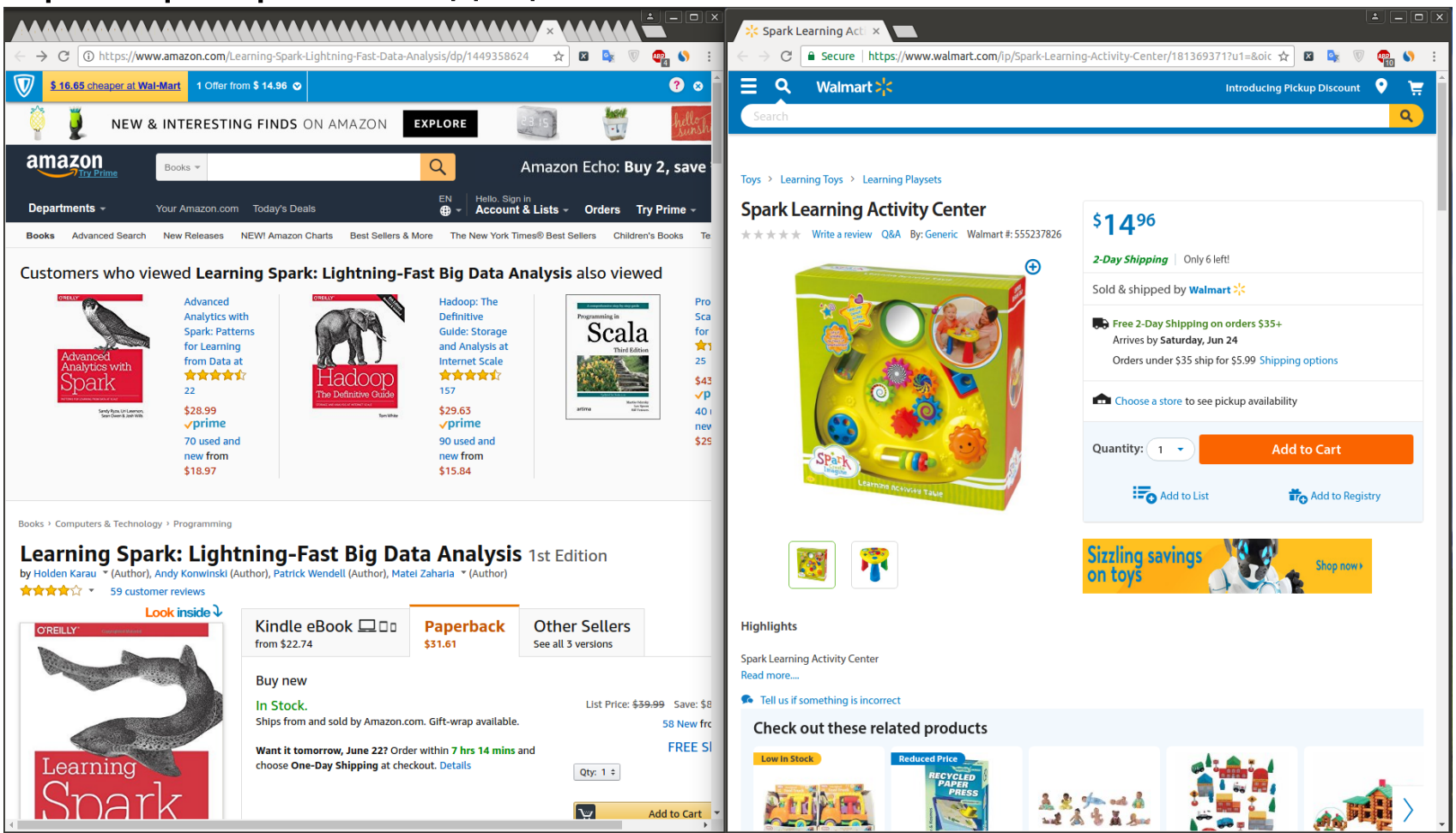
Омонимы – одинаковые по звучанию и написанию, но разные по значению слова, морфемы и другие единицы языка. Выделяют следующие виды омонимии:

- Фонетическая омонимия – совпадение слов только по звучанию: кот – код.
- Графическая омонимия – совпадение слов только по написанию при сохранении различий в звучании: за'мок – замо'к (омографы).
- Морфологические омонимы – совпадение слов, принадлежащих к разным частям речи, в одной или нескольких грамматических формах: печь (глагол) – печь (сущ.) – омоформы.

Паронимы – слова, близкие по звучанию и морфемному строению, но имеющие разный смысл. Надеть (пальто на себя) – одеть (ребенка).

Типы отношений

Разрешение омонимии – серьезная проблема в информационном поиске и рекомендательных системах. Пример – рекомендации плагина Walmart.



Спасибо за внимание!