

Ghi chú của một coder

Vũ Anh

Tháng 01 năm 2018

Mục lục

Mục lục	1
I Khoa học dữ liệu	2
1 Xử lý ngôn ngữ tự nhiên	3
1.1 Introduction to Natural Language Processing	3
1.2 Natural Language Processing Tasks	4
1.3 Natural Language Processing Applications	6
1.4 Spelling Correction	6
1.5 Word Vectors	7
1.6 Conditional Random Fields in Name Entity Recognition	8
1.7 Entity Linking	9
1.8 Chatbot	10
1.8.1 3 loại chatbot	10
1.8.2 Ý tưởng chatbot	11
1.8.3 Một số chatbot	12
Tài liệu	14
Chỉ mục	15
Ghi chú	15

Phần I

Khoa học dữ liệu

Chương 1

Xử lý ngôn ngữ tự nhiên

Bản lưu cũ http://magizbox.com/training/natural_language_processing/site/
Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages.

1.1 Introduction to Natural Language Processing

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages.

NLP is related to the area of human-computer interaction. Many challenges in NLP involve: natural language understanding, enabling computers to derive meaning from human or natural language input; and others involve natural language generation.

The input and output of an NLP system can be either speech or written text. Components of NLP

There are two components of NLP as given

Natural Language Understanding (NLU): this task mapping the given input in natural language into useful representations and analyzing different aspects of the language. Natural Language Generation (NLG): In the process of producing meaningful phrases and sentences in the form of natural language from some internal representation. It involves text planning retrieve the relevant content from knowledge base, sentence planning choose required words, forming meaningful phrases, setting tone of the sentence, text realization map sentence plan into sentence structure. Difficulties

Natural Language has an extremely rich form and structure. It is very ambiguous. There can be different levels of ambiguity

Lexical ambiguity: it is at very primitive level such as word-level. For example, treating the word “board” as noun or verb? Syntax level ambiguity: A sentence be parsed in different ways. For example, “He lifted the beetle with the red cap?” - did he use cap to lift the beetle or he lifted a beetle that had red cap? Referential ambiguity: referring to something using pronouns. For example, Rima went to

Gauri. She said “I am tired”. - Exactly who is tired? One input can mean different meanings. Many inputs can mean the same thing.

1.2 Natural Language Processing Tasks

The analysis of natural language is broken into various board levels such as phonological, morphological, syntactic, semantic, pragmatic and discourse analysis.

Phonological Analysis Phonology is analysis of spoken language. Therefore, it deals with speech recognition and generation. The core task of speech recognition and generation system is to take an acoustic waveform as input and produce as output, a string of words. The phonology is a part of natural language analysis, which deals with it. The area of computational linguistics that deals with speech analysis is computational phonology

Example: Hans Rosling’s shortest TED talk

Original Sound

0:00 / 0:52

Text X means unknown but the world is pretty known it’s seven billion people have seven stones. One billion can save money to fly abroad on holiday every year. One billion can save money to keep a car or buy a car. And then three billion they save money to pay the by be a bicycle or perhaps a two-wheeler. And two billion they are busy saving money to buy shoes. In the future they will get rich and these people we move over here, these people will move over here, we will have two billion more in the world like this and the question is whether the rich people over there are prepared to be integrated in the world with 10 bilions people. Auto generated sound

0:00 / 0:36

Morphological Analysis It is the most elementary phase of NLP. It deals with the word formation. In this phase, individual words are analyzed according to their components called “morphemes”. In addition, non-word taken such as punctuation, etc. are separated from words. Morpheme is basic grammatical building block that makes words.

The study of word structure is refereed to as morphology. In natural language processing, it is done in morphological analysis. The task of breaking a word into its morphemes is called morphological parsing. A morpheme is defined as minimal meaningful unit in a language, which cannot be further broken into smaller units.

Example: word fox consists a single morpheme, as it cannot be further resolved into smaller units. Whereas word cats consists two morphemes, the morpheme “cat” and morpheme “s” indicating plurality.

Here we defined the term meaningful. Though cat can be broken in “c” and “at”, but these do not relate with word “cat” in any sense. Thus word “cat” will be dealt with as minimum meaningful unit.

Morphemes are traditionally divided into two types

(i) “free morphemes”, that are able to act as words in isolation (e.g., “thing”, “permanent”, “local”) (ii) “bound morphemes”, that can operate only as part of other words (e.g., “is” ‘ing’ etc) The morpheme, which forms the center part of the word, is also called “stem”. In English, a word can be made up of one or more morphemes, e.g., word - thing -> stem “think” word - localize -> stem “local”,

suffix “ize” word - denationalize -> prefix “de”, stem “nation”, suffix “al”, “ize” The computational tool to perform morphological parsing is finite state transducer. A transducer performs it by mapping between the two sets of symbols, and a finite state transducer does it with finite automaton. A transducer normally consists of four parts: recognizer, generator, translator, and relator. The output of the transducer becomes a set of morphemes.

Lexical Analysis In this phase of natural language analysis, validity of words according to lexicon is checked. Lexicon stands for dictionary. It is a collection of all possible valid words of language along with their meaning.

In NLP, the first stage of processing input text is to scan each word in sentence and compute (or look-up) all the relevant linguistic information about that word. The lexicon provides the necessary rules and data for carrying out the first stage analysis.

The details of words, like their type (noun, verb and adverb, and other details of nouns and verb, etc.) are checked.

Lexical analysis is dividing the whole chunk of text into paragraphs, sentences, and words.

Syntactic Analysis Syntax refers to the study of formal relationships between words of sentences. In this phase the validity of a sentence according to grammar rules is checked. To perform the syntactic analysis, the knowledge of grammar and parsing is required. Grammar is formal specification of rules allowable in the language, and parsing is a method of analyzing a sentence to determine its structure according to grammar. The most common grammar used for syntactic analysis for natural languages are context free grammar (CFG) also called phase structure grammar and definite clause grammar. These grammars are described in detail in a separate actions.

Syntactic analysis is done using parsing. Two basic parsing techniques are: top-down parsing and bottom-up parsing.

Semantic Analysis In linguistics, semantic analysis is the process of relating syntactic structures, from the levels of phrases, clauses, sentences and paragraphs to the level of the writing as a whole, to their language-independent meanings. It also involves removing features specific to particular linguistic and cultural contexts, to the extent that such a project is possible.

The elements of idiom and figurative speech, being cultural, are often also converted into relatively invariant meanings in semantic analysis. Semantics, although related to pragmatics, is distinct in that the former deals with word or sentence choice in any given context, while pragmatics considers the unique or particular meaning derived from context or tone. To reiterate in different terms, semantics is about universally coded meaning, and pragmatics the meaning encoded in words that is then interpreted by an audience

Discourse Analysis The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.

Topics of discourse analysis include:

The various levels or dimensions of discourse, such as sounds, gestures, syntax, the lexicon, style, rhetoric, meanings, speech acts, moves, strategies, turns, and other aspects of interaction Genres of discourse (various types of discourse in politics, the media, education, science, business, etc.) The relations between text (discourse) and context The relations between discourse and power The relations between discourse and interaction The relations between discourse and cognition

and memory Pragmatic Analysis During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

Sentiment Analysis MetaMind, @RichardSocher

Named Entity Recognition KDD 2015 Tutorial: Automatic Entity Recognition and Typing from Massive Text Corpora - A Phrase and Network Mining Approach

Relationship Extraction AlchemyAPI

1.3 Natural Language Processing Applications

Information Retrieval (IR) Information retrieval (IR) is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on metadata or on full-text (or other content-based) indexing.

Information Extraction (IE) Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most of the cases this activity concerns processing human language texts by means of natural language processing (NLP).

Machine Translation Machine translation, sometimes referred to by the abbreviation MT (not to be confused with computer-aided translation, machine-aided human translation (MAHT) or interactive translation) is a sub-field of computational linguistics that investigates the use of software to translate text or speech from one language to another.

Question Answering (QA) Question answering (QA) is a computer science discipline within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language.

1.4 Spelling Correction

For instance, we may wish to retrieve documents containing the term carrot when the user types the query carot. Google reports (<http://www.google.com/jobs/britney.html>) that the following are all treated as misspellings of the query britney spears: britian spears, britney's spears, brandy spears and prittany spears

We look at two steps to solving this problem: the first based on edit distance and the second based on k-gram overlap. Before getting into the algorithmic details of these methods, we first review how search engines provide spell-correction as part of a user experience.

Implementing spelling correction There are two basic principles underlying most spelling correction algorithms.

Of various alternative correct spellings for a mis-spelled query, choose the nearest one. This demands that we have a notion of nearness or proximity between a pair of queries. When two correctly spelled queries are tied (or nearly tied), select the one that is more common. For instance, grunt and grant both seem equally plausible as corrections for grnt. Then, the algorithm should choose the more common of grunt and grant as the correction. The simplest notion of more

common is to consider the number of occurrences of the term in the collection; thus if grunt occurs more often than grant, it would be the chosen correction. A different notion of more common is employed in many search engines, especially on the web. The idea is to use the correction that is most common among queries typed in by other users. The idea here is that if grunt is typed as a query more often than grant, then it is more likely that the user who typed grnt intended to type the query grunt. Corpus Birkbeck spelling error corpus

References How to Write a Spelling Corrector. Peter Norvig. 2007 Statistical Natural Language Processing in Python. Peter Norvig. 2007 Spelling correction. Introduction to Information Retrieval. 2008

1.5 Word Vectors

Discrete Representation Use a taxonomy like WordNet that has hypernyms (is-a) relationships

```
from nltk.corpus import wordnet as wn
panda = wn.synset("panda.n.01")
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
[Synset('procyonid.n.01'),
Synset('carnivore.n.01'),
Synset('placental.n.01'),
Synset('mammal.n.01'),
Synset('vertebrate.n.01'),
Synset('chordate.n.01'),
Synset('animal.n.01'),
Synset('organism.n.01'),
Synset('living_thing.n.01'),
Synset('w
```

Great as resource but missing nuances, e.g. synonyms: adept, expert, good, practiced, proficient, skillful? Missing new words (impossible to keep up to date): wicked, badass, nifty, crack, ace, wizard, genius, ninja Subjective Requires human labor to create and adapt Hard to compute accurate word similarity Word2Vec Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Word2vec was created by a team of researchers led by Tomas Mikolov at Google. The algorithm has been subsequently analysed and explained by other researchers. Embedding vectors created using the Word2vec algorithm have many advantages compared to earlier algorithms like Latent Semantic Analysis.

Main Idea of Word2Vec

Instead of capturing cooccurrence counts directly,, Predict surrounding words of every word Both are quite similar, see “Glove: Global Vectors for Word Representation” by Pennington et al. (2014) and Levy and Goldberg (2014)... more later. Faster and can easily incorporate a new sentence/document or add a word to the vocabulary. Detail of Word2Vec

Predict surrounding words in a window of length m of every word. Objective function: Maximize the log probability of any context word given the current center word: $J() = \sum_t \sum_{j \in [-m, m]} \log p(w_{t+j} | w_t)$ $J() = \sum_t \sum_{j \in [-m, m]} \log p(w_{t+j} | w_t)$ where θ represents all variables we optimize

Predict surrounding words in a window of length m of every word For $p(w_{t+j} | w_t) p(w_{t-j} | w_t)$

the simplest first formulation is $p(o|c) = \exp(u^T o v_c) / W$ $w = 1 \exp(u^T w v_c)$ $p(o|c) = \exp(u^T o v_c) / W$ where o is the outside (or output) word id, c is the center word id, u and v are “center” and “outside” vectors of o and c

Every word has two vectors! This is essentially “dynamic” logistic regression Linear Relationships in word2vec

These representations are very good at encoding dimensions of similarity!

Analogies testing dimensions of similarity can be solved quite well just by doing vector subtraction in the embedding space Syntactically

xapplexapplesxcarxcarsxfamilyxfamiliesxapplexapplesxcarxcarsxfamilyxfamilies Similarly for verb and adjective morphological forms Semantically (Semeval 2012 task 2)

xshirtxclothingxchairxfurniturexshirtxclothingxchairxfurniture xkingxmanxqueenx-womanxkingxmanxqueenxwoman GloVe Project

Highlights Training Model Overview GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

Pre-trained Model fastText

Pre-trained word vectors for 294 languages, trained on Wikipedia using fast-Text. These vectors in dimension 300 were obtained using the skip-gram model described in Bojanowski et al. (2016) with default parameters.

glove

Pre-trained word vectors. This data is made available under the Public Domain Dedication and License v1.0 whose full text can be found at: <http://www.opendatacommons.org/licenses/pddl>

Language: English

Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, 300d vectors, 822 MB download): glove.6B.zip Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB download): glove.42B.300d.zip Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download): glove.840B.300d.zip Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 25d, 50d, 100d, 200d vectors, 1.42 GB download): glove.twitter.27B.zip word2vec-GoogleNews-vectors

Language: English

Pre-trained Google News corpus (3 billion running words) word vector model (3 million 300-dimension English word vectors).

Word Analogies Test for linear relationships, examined by Mikolov et al. (2014) Suggested Readings Simple Word Vector representations: word2vec, GloVe. cs224d.stanford.edu. Last Accessed: 2017-02-01. FastText and Gensim word embeddings. rare-technologies.com. Last Accessed: 2016-08-31. Distributed Representations of Words and Phrases and their Compositionality. papers.nips.cc. Last Accessed: 2013-12-05. Efficient Estimation of Word Representations in Vector Space. arxiv.org. Last Accessed: 2013-01-16

1.6 Conditional Random Fields in Name Entity Recognition

In this tutorial, I will write about how to using CRF++ to train your data for name entity recognition task.

Environment:

Ubuntu 14.04 Install CRF++ Download CRF++-0.58.tar.gz

Extract CRF++-0.58.tar.gz file
 Navigate to the location of extracted folder through
 Install CRF++ from source
 ./configure make sudo make install ldconfig Congratulations! CRF++ is installed
 crflearnTrainingCRFToTrainACRFusingCRF++, you need 2 things :
 A template file: where you define features to be considered for training A training
 data file: where you have data in CoNLL format crflearn-ttemplatefiletrain_data_filemodel
 crflearn-ttemplatefiletrain.txt modelAbinaryofmodelisproduce.
 To test this model, on a testing data
 crflearn-mmodeltestfile > output.txt
 crflearn-mmodeltest.txt > output.txtReferencesConditionalRandomFields :
 InstallingCRF++onUbuntuConditionalRandomFieldsTrainingandTestingusingCRF++
 +

1.7 Entity Linking

In natural language processing, entity linking, named entity linking (NEL), named entity disambiguation (NED), named entity recognition and disambiguation (NERD) or named entity normalization (NEN) is the task of determining the identity of entities mentioned in text. More precise, it is the task of linking entity mentions to entries in a knowledge base (e.g., DBpedia, Wikipedia)

Entity linking requires a knowledge base containing the entities to which entity mentions can be linked. A popular choice for entity linking on open domain text are knowledge-bases based on Wikipedia, in which each page is regarded as a named entity. NED using Wikipedia entities has been also called wikification (see Wikify! an early entity linking system). A knowledge base may also be induced automatically from training text or manually built.

NED is different from named entity recognition (NER) in that NER identifies the occurrence or mention of a named entity in text but it does not identify which specific entity it is

Examples Example 1:

For example, given the sentence “Paris is the capital of France”, the idea is to determine that “Paris” refers to the city of Paris and not to Paris Hilton or any other entity that could be referred as “Paris”.

Example 2:

Give the sentence “In Second Debate, Donald Trump and Hillary Clinton Spar in Bitter, Personal Terms”, the idea is to determine that “Donald Trump” refer to an American politician, and “Hillary Clinton” refer to 67th United States Secretary of State from 2009 to 2013.

Architecture

Mention detection: Identification of text snippets that can potentially be linked to entities Candidate selection: Generating a set of candidate entities for each mention Disambiguation: Selecting a single entity (or none) for each mention, based on the context Mention detection

Goal: Detect all “linkable” phrases

Challenges:

Recall oriented: Do not miss any entity that should be link Find entity name variants (e.g. “jlo” is name variant of [Jennifer Lopez]) Filter out inappropriate ones (e.g. “new york” matches >2k different entities) COMMON APPROACH

Build a dictionary of entity surface forms entities with all names variants Check all document n-grams against the dictionary the value of n is set typically between 6 and 8 Filter out undesired entities Can be done here or later in the pipeline Examples

Candidate Selection

Goal: Narrow down the space of disambiguation possibilities

Balances between precision and recall (effectiveness vs. efficiency)

Often approached as ranking problem: keeping only candidates above a score/rank threshold for downstream processing.

COMMONNESS Perform the ranking of candidate entities based on their overall popularity, i.e., “most common sense”

Examples

Commonness can be pre-computed and stored in the entity surface form dictionary. Follows a power law with a long tail of extremely unlikely senses; entities at the tail end of distribution can be safely discarded (e.g., 0.001 is sensible threshold)

Disambiguation

Baseline approach: most common sense

Consider additional types of evidence: prior importance of entities and mentions, contextual similarity between the text surrounding the mention and the candidate entity, coherence among all entity linking decisions in the document.

Combine these signals: using supervised learning or graph-based approaches

Optionally perform pruning: reject low confidence or semantically meaning less annotations.

References “Entity Linking”. wikipedia “Entity Linking”. Krisztian Balog, University of Stavanger, 10th Russian Summer School in Information Retrieval. 2016 “An End-to-End Entity Linking Approach for Tweets”. Ikuya Yamada, Hideaki Takeda, Yoshiyasu Takefuji. 2015

1.8 Chatbot

1.8.1 3 loại chatbot

Bài này là dịch từ [bài của một bác ở IBM](#). Nóng hổi vừa thổi vừa dịch.

24/01/2018
Hôm qua vừa nhận kèo cà phê với CEO của Rabiloo. Thấy thú vị quá. Hôm nay hỏi anh Vũ qua qua về chatbot. Hehe

Chatbot hỗ trợ - Support Chatbots

Những con này có xu hướng **nắm rõ về một lĩnh vực**, giống như các kiến thức của một công ty.

Có lẽ Rabiloo, otonhanh, và rất nhiều bot trên facebook thuộc loại này.

Chatbot chức năng - Skill chatbot

Các chatbot chức năng thường là loại **một câu lệnh**, và không cần phải quá chú ý về ngữ cảnh.

Ví dụ, nó có thể thực hiện các câu "Bật đèn lên"

Mấy con bot này có thể ở các nhà thông minh hay các bot trong công nghiệp.

Trợ lý ảo

Các trợ lý ảo có thể kết hợp của hai loại trên. Hoạt động tốt nào biết một chút kiến thức của mỗi lĩnh vực.

Một ví dụ là bạn Siri của Apple.

Kết

Không cần biết loại chatbot bạn muốn xây dựng là gì, điều quan trọng là hãy **đưa cho nó một cuộc sống, một tính cách riêng, khiến nó trở nên hữu ích, và dễ dàng sử dụng.**

Mọi người sử dụng chatbot vì họ muốn có một cách giao tiếp tự nhiên hơn so với những cách trước đó. Có thể đó là công việc đơn giản như bật một chiếc đèn, hay đó là công việc phức tạp như cho vay thế chấp. Mỗi một công việc có những đặc tính cụ thể, chắc chắn chủ bot của bạn tỏa sáng với thiết kế của nó. **Các phương pháp là không thể đếm xuể.**

1.8.2 Ý tưởng chatbot

Mình muốn chatbot giao tiếp bằng ngôn ngữ tự nhiên chứ không theo kiểu mệnh lệnh. *cười -> ra chuyện cười*. Thế thì không khác gì search engine.

Mà sao không có quan bot nào khai thác tâm trạng, thông tin của người dùng. Để tìm ra nhu cầu của họ nhỉ?

Vài chức năng hữu ích

- Hỏi đáp thông tin cá nhân của bot. Nhiều bot fail bởi trò đơn giản này.
- Yêu đương an ủi tán tỉnh
- Dự báo thời tiết. Lấy thông tin địa điểm người dùng. Bình luận về thời tiết hiện tại
- Tìm ảnh gái xinh, đọc truyện cười
- Khả năng học

Chán chẳng cần làm

- Tra mã số karaoke

1.8.3 Một số chatbot

Chatbot tiếng anh

1. [Mitsuku \(web\)](#) (2000-). Chiến thắng Loebner Prize vào năm 2016-2017
2. [Rose \(web\)](#). Chiến thắng Loebner Prize vào năm 2014-2015
3. [Cleverbot](#) (1997)
4. [ELIZA](#) (1964-1966) tại MIT bởi Joseph Weizenbaum
5. [poncho \(messenger\)](#). Con này chỉ chuyên về thời tiết
6. [Melody \(Baidu\)](#). Chuyên tư vấn về bác sỹ
7. [Do Not Pay](#) (2017). Chatbot về tư vấn luật
8. [meditatebot \(messenger\)](#) - Hướng dẫn thiền
9. [bots duolingo \(ios\)](#) - Bot hướng dẫn học ngoại ngữ

Nền tảng tiếng Anh

- [api.ai](#)
- [DialogFlow](#)

Cuộc thi tiếng Anh

[Loebner Prize](#). Được tổ chức lần đầu vào năm 1990. Mục tiêu là kiểm tra xem chương trình có vượt qua được Turning Test hay không.

Chatbot tiếng việt

1. [troly.bedieu \(messenger\)](#) (2016) Có mấy chức năng. Tán ngẫu. Tìm ảnh gái xinh. Ổn phết
2. [Bob \(skype\)](#) Cũng khá vui đấy
3. [Người Bạn Tốt Miki](#) (2016). Xìt rồi
4. [hana.ai](#). Nghe nói có vẻ khủng
5. [Simsimi](#). Thấy bảo "con" chatbot này bậy lắm. Không biết có thật không?
6. [Sumichat](#) Hỗ trợ cả tiếng Việt đây
7. [VisualFriend](#) (2007) bởi HungCode. Nghe nói là thần thánh lắm nhưng chưa kiểm chứng.

Nền tảng tiếng việt

1. [hekate.ai](#) (2017). 66 triệu doanh nghiệp. 1.2 tỷ người. 68 tỷ message mỗi ngày
2. [harafunnel.com](#)

Cuộc thi trên nền tảng tiếng Việt

[fpt.ai: bot of the year](https://fpt.ai/bot-of-the-year)

Đề bài: Xây dựng Chatbot để nâng cao trải nghiệm của người dùng cá nhân và doanh nghiệp

Thời gian: 22/11/2017 – 30/07/2018

Đối tượng: Cá nhân và doanh nghiệp quan tâm đến lĩnh vực Chatbot trên khắp Việt Nam.

Tài liệu tham khảo

Ghi chú

■ 24/01/2018 Hôm qua vừa nhận kèo cà phê với CEO của Rabiloo. Thấy
thú vị quá. Hôm nay hỏi anh Vũ qua qua về chatbot. Hehe 10