Meeting Overview

1. Topics analysis→ information retrieval

2. Hypotheses and questions

From
topic analysis
to
information retrieval

GOAL: Topic → Information

Topic Analysis (treat it as black box function)

Topic 1: 0.574*"flight" + 0.292*"seat" + 0.279*"air" + 0.262*"canada" + 0.144*"get" + 0.129*"service" + 0.129*"time" + 0.126*"fly" + 0.125*"hour" + 0.117*"toronto"

Topic 2: -0.744*"seat" + 0.380*"flight" + -0.144*"economy" + -0.136*"business" + 0.112*"hour" + -0.109*"new" + -0.105*"class" + 0.102*"air" + 0.094*"canada" + 0.090*"delay"

Topic 3: 0.542*"canada" + 0.541*"air" + -0.446*"flight" + -0.191*"good" + -0.095*"cabin" + 0.089*"fly" + -0.084*"food" + -0.084*"attendant" + -0.077*"economy" + 0.075*"passenger"

Topic 4: -0.294*"get" + 0.261*"good" + -0.258*"seat" + 0.193*"service" + 0.191*"canada" + 0.187*"food" + 0.186*"air" + -0.184*"toronto" + -0.184*"tell" + 0.175*"class"

Topic 5: 0.424*"flight" + 0.252*"seat" + -0.185*"passenger" + -0.169*"get" + -0.158*"time" + -0.156*"airline" + -0.155*"check" + -0.151*"board" + -0.148*"staff" + -0.147*"service"

Information retrieved described

Flight: e.g. had delay ?

Seat: e.g. uncomfortable ?

Service: e.g. good cust. Service ?

Delay: e.g. yes/no ?

Food: e.g. cold / delicious ?

Economy class: e.g. cheap ?

Business class: e.g. expensive ?

Staff: e.g. friendly ?

Passengers: e.g. noisy ?

Introduction to word2vec

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

Word2Vec

- What is it ?

A continuous bag-of-words and skip-gram architectures for computing vector representations of words

- How it works ?

word2vec takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words. The resulting word vector file can be used as features in many natural language processing and machine learning applications.

- So basically it finds distances between words in the vector space

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Other tools:

- Multiword phrase detection
- Similarity detection
- Summary (prioritized sentences)
- Summary keywords
- Part-of-speech syntactic parser
- doc2vec

Other concerns

A.) Perform topic analysis on : all airline reviews **vs** each airline company individually

B.) Topic words validation: topic words **vs** summary keywords. Does it make sense ?

C.) How to cluster topic words ?

1. Find a way to cluster words/topics
2. When you see overlap, you found correlation

D.) Adjective analysis: Adjectives describe the quality/ property of actions/ objects. How could we do a qualitative analysis with adjectives, to find out **HOW** is something?

E.) Filter out garbage: How to identify topic words that are unrelated / make no sense to put in a summary