Artificial Intelligence
EDA132
Lecture 14.1: Information Extraction

Pierre Nugues

Lund University
Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

March 3, 2017

Text Categorization

The objective is to determine the type of a text with a set of predefined categories, for instance: {spam, no spam}
The Reuters corpus contains 800,00 economic newswires (`http://trec.nist.gov/data/reuters/reuters.html`)
Each newswire is manually annotated with a topic selected from a set of 103 predefined topics, for example:

C11: STRATEGY/PLANS,

C12: LEGAL/JUDICIAL,

C13: REGULATION/POLICY,

C14: SHARE LISTINGS

etc.

Text Representation

Most categorizers use the **bag-of-word** technique that represents each document as a vector of words.
The vector parameters denote the presence or absence of a word.
The documents:

D1: Chrysler plans new investment in Latin America.

D2: Chrysler plans major investments in Mexico.

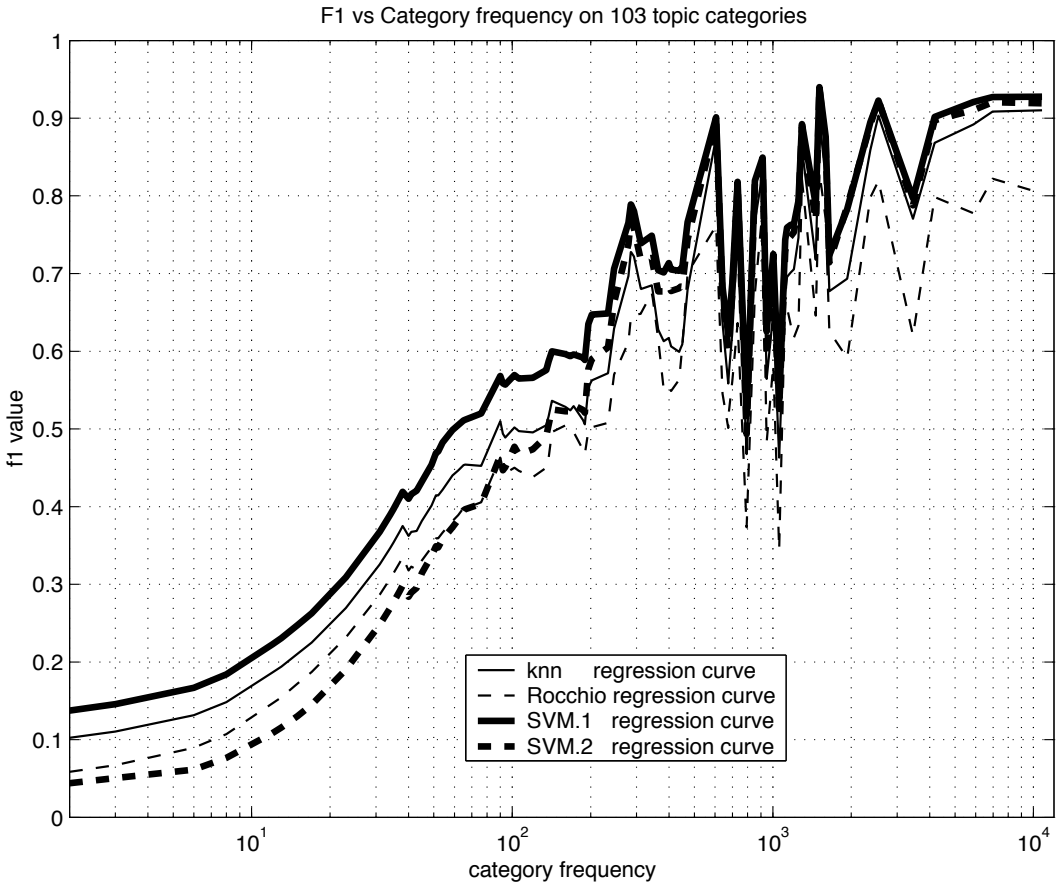
are represented as:

D\W	chrysler	plan	new	major	investment	latin	america	mexico
1	1	1	1	0	1	1	1	0
2	1	1	0	1	1	0	0	1

We can use supervised learning, where the classes are the categories and the features, the word vectors.

Algorithms for Text Categorization

Support vector machines proved to have the best performance.



David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li, RCV1: A New Benchmark Collection for Text Categorization Research, *Journal of Machine Learning Research* 5 (2004) 361-397.

Information Retrieval

Astronomic number of available documents
Search engines – Google, Yahoo – are examples of tools to retrieve information on the web
Usually, we have:

- A document collection
- A query
- A result consisting of a set of documents

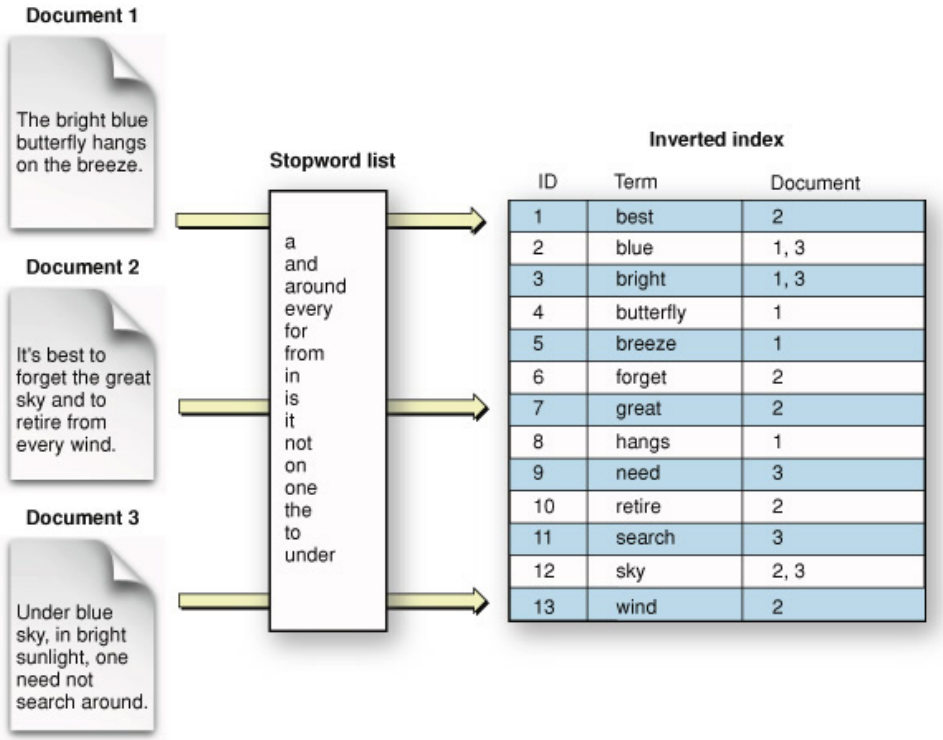
The simplest technique is to use a Boolean formula of conjunctions and disjunctions that will return the documents satisfying it.

The Vector Space Model

The vector space model represents a document in word space:

Documents \Words	w_1	w_2	w_3	...	w_m
D_1	$C(w_1, D_1)$	$C(w_2, D_1)$	$C(w_3, D_1)$	...	$C(w_m, D_1)$
D_2	$C(w_1, D_2)$	$C(w_2, D_2)$	$C(w_3, D_2)$	...	$C(w_m, D_2)$
...					
D_n	$C(w_1, D_n)$	$C(w_2, D_n)$	$C(w_3, D_n)$	...	$C(w_m, D_n)$

Inverted Index (Source Apple)



http://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/SearchKitConcepts/index.html

Lucene is an outstanding program for document indexing and retrieval:

http://lucene.apache.org

Giving a Weight

Word clouds give visual weights to words

Image: Courtesy of Jonas Wisbrant

$TF \times IDF$

The frequency alone might be misleading

Document coordinates are in fact $tf \times idf$: Term frequency by inverted document frequency.

Term frequency $tf_{i,j}$: frequency of term j in document i

Inverted document frequency: $idf_j = \log(\frac{N}{n_j})$

Document Similarity

Documents are vectors where coordinates could be the count of each word:
$\vec{d} = (C(w_1), C(w_2), C(w_3), ..., C(w_n))$
The similarity of documents is their cosine:

$$\cos(\vec{q}, \vec{d}) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2} \sqrt{\sum_{i=1}^{n} d_i^2}}.$$

Evaluation

The Message Understanding Conferences have introduced a metric to evaluate the performance of information extraction systems using three figures.
They are borrowed them from library science

	Relevant documents	Irrelevant documents
Retrieved	A	B
Not retrieved	C	D

Recall, Precision, and the F-Measure

Recall measures how much relevant information the system has retrieved.

$$\text{Recall} = \frac{A}{A \cup C}.$$

Precision is the accuracy of what has been returned

$$\text{Precision} = \frac{A}{A \cup B}.$$

Recall and precision are combined into the **F-measure**, which is defined as the harmonic mean of both numbers:

$$F = \frac{2PR}{P + R}.$$

Implementation Details

Very frequent words (stop words) can be removed
Words can be stemmed or lemmatized, for instance *table*, *tables*, *tabled*, *tabling* would have the same representation
Search can be extended to synonyms
Some systems use spell checkers

Google's PageRank

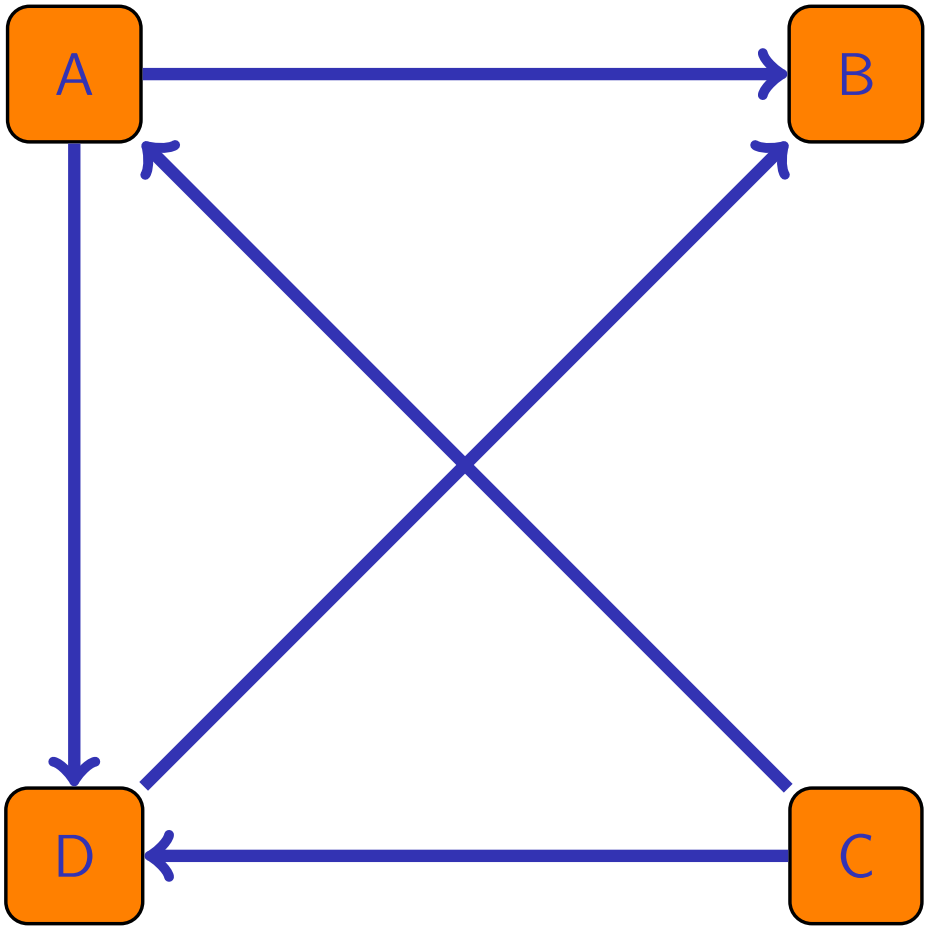
Google's PageRank algorithm does not use word frequencies, but the page popularity through the "backlinks", the links pointing to a page.
Each backlink has a specific weight, which is related to the rank of the page it comes from.
The page rank is defined as the sum of the weights of all its backlinks:

$$PR(p_j) = \frac{1-d}{N} + d \sum_{p_i \in Ref(p_j)} \frac{PR(p_i)}{C(p_i)}$$

The importance of a page is spread through its forward links and contributes to the popularity of the pages it points to.
The weight of each of these forward links is the page rank divided by the count of the outgoing links.
The ranks are propagated in a document collection until they converge:

$$PR(p_j, t+1) = \frac{1-d}{N} + d \sum_{p_i \in Ref(p_j)} \frac{PR(p_i, t)}{C(p_i)}$$

Pagerank Example



$$d = 0.85; \frac{1-d}{N} = \frac{0.15}{4} = 0.0375$$

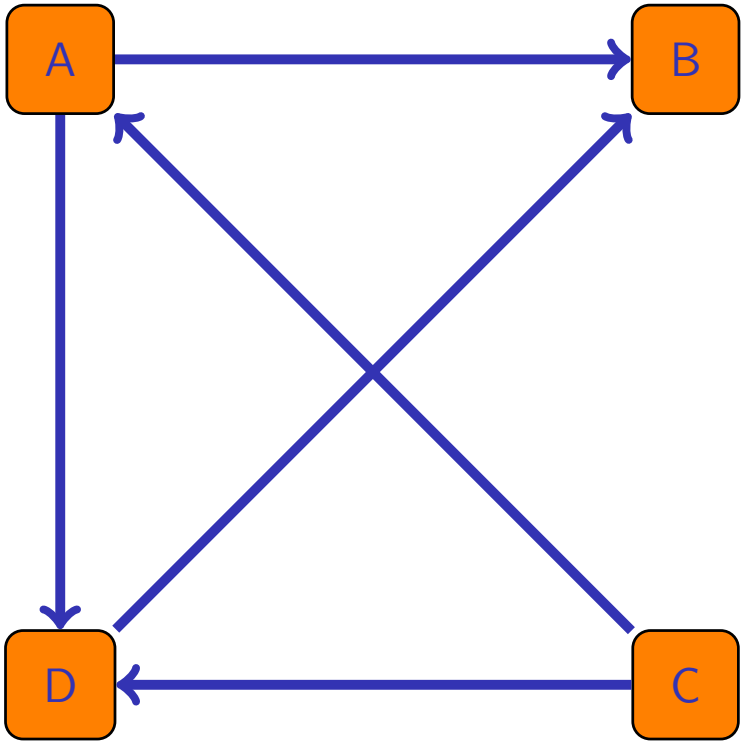
A	B	C	D
1	1	1	1

Table: Initial pageranks

$\rightarrow$	A	B	C	D
A	0	1	0	1
B	0	0	0	0
C	1	0	0	1
D	0	1	0	0

Table: Links

Pagerank Example

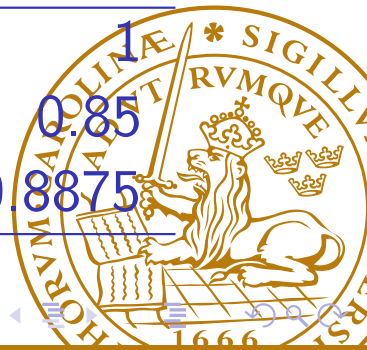


$\rightarrow$	A	B	C	D
A	0	0.5	0	0.5
B	0	0	0	0
C	0.5	0	0	0.5
D	0	1	0	0
Sum	0.5	1.5	0	1

Table: Initial weighted links

	A	B	C	D
Sum	0.5	1.5	0	1
Damped $= 0.85 \times$ Sum	0.425	1.275	0	0.85
PR $= 0.0375 +$ Damped	0.4625	1.3125	0.0375	0.8875

Table: Pageranks after one iteration



Message Understanding Conferences

The Message Understanding Conferences (MUCs) measure the performance of information extraction systems.
They are competitions organized by an agency of the US department of defense, the DARPA
The competitions have been held regularly until MUC-7 in 1997.
The performances improved dramatically in the beginning and stabilized then.
MUCs are divided into a set of tasks that have been changing over time.
The most basic task is to extract people and company names.
The most challenging one is referred to as information extraction.

Information Extraction

Information extraction consists of:

- The analysis of pieces of text ranging from one to two pages,
- The identification of entities or events of a specified type,
- The filling of a pre-defined template with relevant information from the text.

Information extraction then transforms free texts into tabulated information.

An Example

San Salvador, 19 Apr 89 (ACAN-EFE) – [TEXT] Salvadoran President-elect Alfredo Cristiani condemned the terrorist killing of Attorney General Roberto Garcia Alvarado and accused the Farabundo Marti National Liberation Front (FMLN) of the crime...
Garcia Alvarado, 56, was killed when a bomb placed by urban guerrillas on his vehicle exploded as it came to a halt at an intersection in downtown San Salvador...
Vice President-elect Francisco Merino said that when the attorney general's car stopped at a light on a street in downtown San Salvador, an individual placed a bomb on the roof of the armored vehicle...
According to the police and Garcia Alvarado's driver, who escaped unscathed, the attorney general was traveling with two bodyguards. One of them was injured.

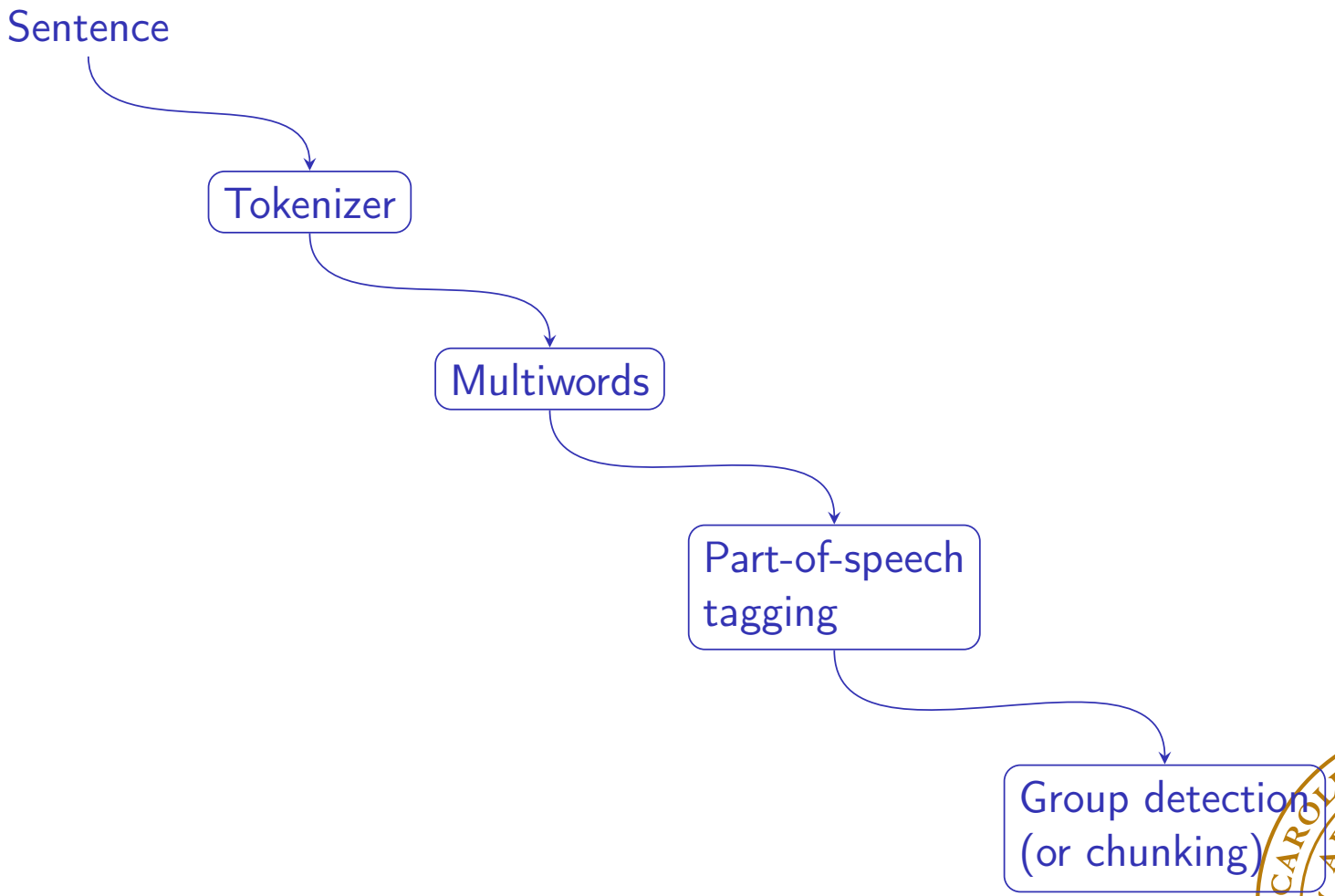
The Template

Template slots	Information extracted from the text
Incident: Date	19 Apr 89
Incident: Location	El Salvador: San Salvador (city)
Incident: Type	Bombing
Perpetrator: Individual ID	*urban guerrillas*
Perpetrator: Organization ID	*FMLN*
Perpetrator: Organization confidence	Suspected or accused by authorities: *FMLN*
Physical target: Description	*vehicle*
Physical target: Effect	Some damage: *vehicle*
Human target: Name	*Roberto Garcia Alvarado*
Human target: Description	*Attorney general*: *Roberto Garcia Alvarado* *driver* *bodyguards*
Human target: Effect	Death: *Roberto Garcia Alvarado* No injury: *driver* Injury: *bodyguards*

FASTUS

The FASTUS system has been designed at the Stanford Research Institute to extract information from free-running text
FASTUS uses partial parsers that are organized as a cascade of finite-state automata.
It includes a tokenizer, a multiword detector, and a group detector as first layers.
Verb groups are tagged with active, passive, gerund, and infinitive features.
Then FASTUS combines some groups into more complex phrases and uses extraction patterns to fill the template slots.
See
`http://www.ai.sri.com/natural-language/projects/fastus.html`

FASTUS' Architecture



Probabilistic Models for Information Extraction

It is possible to use statistical tagging techniques to carry out information extraction.
An example with three tapes corresponding to the text (input), speaker, and date (both output).
(From the textbook, Stuart Russell and Peter Norvig, *Artificial Intelligence*, 3rd ed., 2010, page 876.)

There	will	be	a	seminar	by	Andrew	McCallum	on	Friday
–	–	–	–	PRE	PRE	TARGET	TARGET	POST	–
–	–	–	–	–	–	–	–	PRE	TARGET

The speaker and date tapes are tagged by two separate hidden Markov models.
The procedure is similar to that of part-of-speech tagging.

Conditional Random Fields

To carry out named entity tagging, time or location extraction, it is possible to use discriminative models such as support vector machines. Conditional random fields are tools that take into account sequences. Let X_1^N be the words and Y_1^N, the tags:

$$P(Y_1^N|X_1^N) = \alpha e^{\sum_{i=1}^{N} F(Y_{i-1}, Y_i, X_1^N, i)}$$

F are feature functions:

$$F(Y_{i-1}, Y_i, X_1^N, i) = \sum_k \lambda_k f_k(Y_{i-1}, Y_i, X_1^N, i)$$

where:

$$f_1(Y_{i-1}, Y_i, X_1^N, i) = \begin{cases} 1 & Y_i = \text{SPEAKER and } X_i = \text{Andrew,} \\ 0 & \text{otherwise.} \end{cases}$$

(From the textbook, Stuart Russell and Peter Norvig, *Artificial Intelligence*, 3rd ed., 2010, page 879.)

Question Answering

Question parsing and classification: Syntactic parsing, entity recognition, answer classification

Document retrieval. Extraction and ranking of passages: Indexing, vector space model.

Extraction and ranking of answers: Answer parsing, entity recognition