8. Information Extraction 1

8.1. Explain what IE is

Information extraction (IE) is the automatic identification of selected types of entities, relations, or events in free text.

8.2. Explain the basics of how a rule based named-entity tagger works

- Large number of (finite state) patterns
- Each pattern captures and classifies some subset of named entities

  Example NERC (named entity Recognition) patterns

  [capitalized-word]+['Corp.'] ⇒ Organization
  ['Mr.'][capitalized-word]+ ⇒ Person
  [in|at|on][capitalized-word]+ ⇒ Location

  Additional rules to handle exceptions ("in Putin's mind" - location?)

  - Use gazetteers, i.e., word lists for each of the NERC categories

    [cap-word-names-gazetteer]+[cap-word-surnames-gazetteer]+

    - Personal names: Aaliyah, Aaron, Abbey
    - Surnames: Abbott, Abney, Abraham
    - Organizations: Abbott Laboratories, Abercrombie & Fitch

8.3. List the most often used named entity types

- Entity names (ENAMEX) – Person, Organization, Location
- Temporal expressions (TIMEX) – Date, Time
- Quantities (NUMEX) – Monetary value, Percentage

8.4. + 8.5. Explain the BIO annotation scheme. Given a sentence annotated with named entities, translate the annotations into BIO tags

- B – Begins a named entity (i.e., first NE token)
- I – Inside a named entity (i.e., second and subsequent NE tokens)
- O – Outside of a named entity (i.e., token is not part of any NE)

  Example:

  Barcelona's/B-Org draw/O with/O Atletico/B-Org Madrid/I-Org at/O Camp/B-Loc Nou/I-Loc was/O not/O expected/O, says/O British/B-Org Broadcast/I-Org Channel's/I-Org football expert Andy/B-Per West/I-Per

8.6. Explain the differences between classification-based and sequence labeling-based approaches to NER and explain the advantage of the latter over the former

- Classification

  Naive Bayes, SVM, Logistic regression, . . .
  Cannot use labels from both token sides as features

- Sequence labeling

  Hidden Markov Models (HMM) and Conditional Random Fields (CRF)
  Model joint probability of a sequence of labels

- Answer: Sequence models predict B-I-O labels on sentence level. Possible to have differing labels for the same named entity at the document level.

8.7. Describe the most important/often used features for NER models

- Linguistic features – word, lemma, POS-tag, morpho-syntactic descriptors, word shape, sentence start, capitalization, acronym
- Gazetteer features – is gazetteer entry, starts gazetteer entry, is inside of gazetteer entry (for all gazetteers)

8.8. Explain what document-level consistency is and how it's useful for NER, and provide an example

- Two-level CRF (soft constraints)
  - Second CRF labeler using global features, which are computed on the labels produced by the first CRF
  - E.g., the most frequent label for the lemma, the most frequent NE label the most frequent superentity label (e.g., "Department of Public Health" is a superentity of "Department")
- Postprocessing rules (hard constraints)
  - Adding rules on top of supervised model
  - E.g., predict for all instances of the same entity the same label (the most frequent label assigned by the CRF)

8.9. Explain briefly how to evaluate NER using both "lenient" and "strict" matching, and demonstrate on an example

- Lenient evaluation (aka MUC evaluation)
  - The same NE type and extent overlap with some gold NE annotation
- Strict evaluation (aka Exact evaluation)
  - The same NE type and exactly the same extent as some gold NE annotation

  NERC evaluation example
  - Gold: "**The Faculty of Electrical Engineering and Computing** issued an. . . "
  - A1: "**The Faculty of Electrical Engineering** and Computing issued an. . . "
  - A2: "**The Faculty of Electrical Engineering and Computing** issued an. . . "

8.10. Explain the concepts of entity mentions, coreferent mentions, and a coreference chain

- Entity mentions are phrases in text that denote entities from the real-world
- Coreferent mentions denote the same entity from the real word
- Coreference chain is a set of all mentions within a document or a collection of documents that denote the same real-world entity

  Coreference resolution is often confused with anaphora resolution, but there is a significant difference
  - Two mentions are coreferring to each other if they denote the same real-world entity
  - Mention A is an anaphoric antecedent of mention B iff A is required for the interpretation of B

8.11. Explain the basic idea behind rule-based coreference resolution

- Linguistically-motivated rules
- Rules based on domain knowledge

  Hobbs' algorithm
    - Within-sentence pronoun resolution algorithm
    - Constraints on pronouns on the constituency syntactic parse of the sentence

  Heuristics based on centering theory
    - Centering theory – inference load on the reader is lower when mentions of the same entities occupy the same grammatical roles

      Centering example
        1. Johnny really goofs around sometimes.
        2. He was excited about trying out his new sailboat.
        3. He wanted Tony to join him on a sailing expedition.
        4. He called him at 6 AM.
        5. He was sick and furious at being woken up so early.

8.12. Explain the idea behind the mention-pair-based ML approach to coreference resolution

- The core mention-pair model is a classifier that determines whether two mentions are coreferent
- Downsides of the binary classification approach:
    1. Transitivity property of the coreference relation is not enforced. A separate clustering algorithm is needed
    2. Creating training instances. One training instance for each pair of mentions in the document? Vast majority of training instances are negative. Additional training instance creation method is needed.
- A mention-pair model must define the following four components:
    1. Training instance creation method
        a. Typically heuristics aiming to reduce number of negative pairs.
        b. Some employ binary classifier to decide should the pair of mentions be a training instance.
    2. Linguistic features for representing an instance
        a. String matching features (e.g., exact match, head noun match, minimum edit distance)
        b. Syntactic features are computed on a syntactic parse tree
        c. Semantic features (e.g., selectional preference, WordNet-based noun similarity)
        d. Discourse-based features (e.g., distances, grammatical prominence)
    3. Learning algorithm, i.e., pairwise coreference classifier
        a. Non-parametric models in early research – decision trees, rule learners, and memory-based learners
        b. Statistical learners in recent research – log. regression, SVM

4. Clustering algorithm for constructing a coreference partition
    a. Closest-first clustering – mention assigned to the cluster of the closest preceding mention classified as coreferent
    b. Best-first clustering – mention assigned to the cluster of the most confident preceding mention
    c. Graph partitioning algorithms applied on a weighted undirected graph (vertices are mentions, weights of edges are classification confidences)

8.13. Explain the basic idea behind the entity-mention-based ML approach to coreference resolution

- Addresses the expressiveness problem of the mention-pair models E.g., “Mr.Clinton”, “Clinton”, “she”
- Main idea: Classify whether a mention to be resolved is coreferent with a preceding, possibly partially formed cluster
    - Comparison with existing clusters instead of individual mentions
- Each training instance consists of a mention and a cluster
    - We need cluster-level features
    - Often computed as aggregation (All, Any, Most) of mention-level pair features. E.g., mention agrees in gender with the cluster if it agrees in gender with All mentions of the cluster
- Despite expressiveness, entity-mention models do not outperform mention-pair models
    - Developing good cluster-level features is hard
    - Aggregation not obvious for most mention-pair features (e.g., distances, syntactic features)

8.14. Describe how the B-CUBED metric for evaluating coreference resolution works

- B^3 (B-CUBED) metric

  Precision and recall are computed for each mention

$$precision(m_i) = \frac{|cluster_{auto}(m_i) \cap cluster_{gold}(m_i)|}{|cluster_{auto}(m_i)|}$$

$$recall(m_i) = \frac{|cluster_{auto}(m_i) \cap cluster_{gold}(m_i)|}{|cluster_{gold}(m_i)|}$$

  Overall precision and recall are averaged over all mentions

  B^3 score = harmonic mean of the overall precision and recall

8.15. Define the task of entity linking

- Entity linking is a task of associating mentions of an entity in text to an entry representing that entity in a knowledge base. Not a trivial task, primarily due to ambiguity (the same string may refer to more than one entity).

  Coref. resolution captures different mentions of the same entity.
  Entity linking often needs to disambiguate between different entities with the same surface form in text

8.16. Explain the basic idea of performing entity linking

- Entity linking is essentially an information retrieval task
  - An entity mention (with the originating document) is a query
  - Knowledge base articles constitute the document collection
- Retrieval models based on semantic comparison between the context of the mention (i.e., originating text) and the knowledge base articles

8.17. Define the task of relation extraction

- Relation Extraction (RE) is the task of recognizing the assertion of a particular relationship between two or more entities in text

  Relation extraction example:

  > Located in Pittsburgh, Carnegie-Mellon University is one of the leading U.S. technical universities alongside Stanford, Berkeley, and MIT
  > R1: Located-in(Carnegie-Mellon University, Pittsburgh)
  > R2: Peers(Carnegie-Mellon University, Stanford, Berkeley, MIT)

8.18. Describe briefly the main ideas behind supervised relation extraction

- Relation extraction as a classification task
- Approaches: feature-based, kernel-based
- Evaluation using precision, recall, and F1-score

9. Information extraction 2

9.1. Explain what an event is and what an event mention is in text

- Loose definition: Event is something that happens or occurs
- Event extraction is the task of extracting information about actions and activities, their participants and circumstances from text
- Provide answers to “Wh-questions” - Who did what to whom, where and when?"
- Events are inherently temporal
- Event extraction typically coupled with temporal information extraction tasks

9.2. List the steps in an event extraction pipeline

1. Event anchor extraction - Anchor is the word of phrase bearing the core meaning of the action/activity
2. Event argument extraction - Arguments denote the participants and circumstances (e.g., time, location, manner)

9.3. Explain in brief how to extract event anchors

- typically performed in a supervised fashion
  - Anchor identification - detects phrases that are event anchors
  - Event type classification - assigns a semantic event type to event
- Features
  - Lexical features
  - POS features (event anchors are most often verbs)
  - Syntactic features
  - Semantic features (e.g., WordNet-based)

9.4. Explain in brief how to determine the type of an event

- By making two subtasks in event argument extraction
  - Argument identification - recognize phrases which denote participants or circumstances of events
  - Argument type classification - assigns the exact argument type to event arguments (AGENT, TARGET, TIME, LOCATION, MANNER)
- Two types of methods
  - Direct classification
  - Methods based on frame semantics

9.5. Describe the idea of approaches for event argument extraction based on direct classification

- Each syntactic argument of the event anchor is an argument candidate
- Classifier determines whether the candidate is indeed an argument
- Classifiers determines the semantic role of an argument

  There are two types of approaches:
  - Supervised classification
  - Rule-based classification

9.6. Explain the idea of approaches for event argument extraction based on frame semantics

- Frame identification
  - Given event anchor (and context), determine an anchor
  - A task similar to word sense disambiguation
  - Frame defines the set of argument roles
  - Supervised and unsupervised models
- Argument identification
  - Determine which syntactic argument of the anchor are semantic arguments of a frame
  - Typically performed in a supervised fashion
- Assignments of semantic roles
  - Semantic roles of the frame assigned to event arguments
  - Supervised and unsupervised models

9.7. Describe what event coreference resolution is and give a basic idea on how to use supervised ML to solve this problem

- Event coreference resolution is finding out which event mentions denote the same real world event.
- Supervised approaches are:
    - Features based on comparison of the two event mentions
    - Semantic similarity of event anchors
    - Semantic similarity of arguments with the same roles

9.8. Explain what a temporal expression is and provide an example

- Temporal expression is expression denoting temporal relations between events and times.
- Example:
    - "in 2006, a poll on NSA surveillance suggested that 51% found NSA surveillance acceptable while 47% found it unacceptable. Shortly after the Snowden disclosures began, public opinion was equally divided about the Section 215 program. And just a few weeks ago, a Pew Research poll from last month found public opinion pretty evenly divided again."

9.9. Explain the difference between the temporal expression identification and temporal expression normalization tasks

- Normalization of temporal expressions aims to assign precise temporal coordinates to temporal expressions

9.10. Explain what temporal expression identification is

- Recognizing the extent of temporal expression in text

9.11. Describe what temporal relation extraction is

- Temporal relation extraction is process of extracting temporal relations between pairs of events and events and temporal expressions

9.12. Explain what a collocation is, and provide examples in English or Croatian

- Collocation is a syntactic and semantic unit whose exact and unambiguous meaning or connotation cannot be derived directly from the meaning or connotation of its components

9.13. Briefly explain how the main three steps constituting a typical collocation extraction pipeline work

- Candidate extraction
    - Extracting candidates from a large corpus using part-of-speech patterns, frequencies, and other heuristics
- Computation of lexical association measures
    - Measures of lexical association between words constituting a candidate collocation expression
- Ranking or classification
    - Ranking candidates according to the assigned lexical association score
    - Supervised classification of candidates as collocations (or not) with lexical association scores as features

9.14. Explain and exemplify the difference between collocations and keyphrases

- Collocation refers to the way in which some words regularly occur together
  - Example: Do homework; Make the bed; A golden opportunity; Take a risk; A faint smell
- Keyphrases are a form of expression that has taken on a more specific meaning than the words themselves. It is the standard way of expressing a concept or an idea; it is something we ordinarily say in certain situations.
  - Example: Pleased to meet you; All of a sudden; On the other hand; More trouble than it's worth; Neither here nor there

9.15. Explain the difference between supervised and unsupervised approaches to keyphrase extraction

- Supervised keyphrase extraction
  - Keyphrase extraction cast as a classification problem
  - Classifying expressions into keyphrases and non-keyphrases
  - Dataset of documents manually annotated with keyphrases is required
- Unsupervised keyphrase extraction
  - Keyphrase extraction cast as a clustering problem
  - No manually annotated documents are required
  - Typically lower performance than with supervised approaches

9.16. Explain the main steps typically involved in a supervised keyphrase extraction algorithm

- Candidate extraction
  - Most often based on heuristics such as part-of-speech patterns, e.g., economic crisis vs. crisis in
- Candidate feature extraction (or scoring)
  - Features like TF-IDF score, position in the document, length, etc.
- Classification with a supervised machine learning algorithm
  - Learning on the manually annotated dataset
  - Algorithms range from Naive Bayes and SVM to genetic programming

9.17. Explain the main steps typically involved in an unsupervised keyphrase extraction algorithm

- Word clustering
  - Word vectors over a collection of documents, where elements are, e.g., TF-IDF scores
  - Clustering words using cosine between vectors as similarity
  - Aim: Obtain topically coherent word clusters

- Tagging documents with clusters
  - For each document D and each word cluster C, a document-cluster score S(D,C) is computed:

        - $S(D, C) = \sum_{w \in D \cap C} f(w, D)^{\alpha} \cdot s(w, C)^{\beta}$

          $f(w, D)$ − frequency of w in D

          $s(w, C)$ − average similarity of w with all other words in C
    - Cluster C that has the highest $S(D, C)$ score is chosen
- Keyword selection
    - Keywords selected from the intersection of words from the document D and cluster C that was chosen in previous step
    - Ranking words from the intersection according to the score $f(w, D)^{\alpha} * s(w, C)^{\beta}$
    - Two selection methods
        - Select n top-scored words
        - Select all words with the score higher than some threshold

10. Semantic Textual Similarity, Summarization, and Simplification

10.1. Explain what semantic textual similarity is and provide examples of semantically similar and semantically dissimilar texts

- Content similarity between texts
- Examples (first two are similar):
    - I called to the large dog that sat on the bank.
    - A huge dog was by the river, I yelled its name.
    - Police were called to the bank after a large amount of money had been stolen

10.2. Describe the issues with simple word overlap or VSM (Vector Space Model)-based approaches, and describe possible solutions for these issues

- Word overlaps and VSM have issues measuring word similarity because they can capture only exact word matches between texts. We can solve this issue by:
    1. Assign standard weighted vectors to documents
    2. Transform standard weighted vectors into lower dimensional semantic vectors
    3. Compare documents by comparing their semantic vectors

10.3. Describe in short the supervised ML approach to semantic textual similarity

- Prerequisite – a dataset containing: Pairs of texts and a human rating of similarity R_H for each pair (typically on a 0–5 scale).
- Features of the best-performing model:
    - String matching: Word/N-gram overlap, skip n-gram overlap
    - Latent semantics: Greedy word alignment overlap, LSA similarity
    - Syntactic: Syntactic role similarities, syntactic dependencies overlap
    - IE-based: Named entities match, number overlap

10.4. Explain how semantic textual similarity systems are evaluated

- Semantic textual similarity is evaluated so that: we have human similarity ratings R_H for each pair in the dataset, and the system provides us a numerical similarity rating R_S for each pair. Then we compute the correlation coefficient between R_H and R_S.

10.5. Explain what authorship attribution is

- Task: determine if a text was written by a specific author

10.6. Briefly explain how a supervised machine learning approach to authorship attribution works

- Machine learning for authorship attribution uses a dataset of texts manually labeled with authors. It's features are:
  - word - length in letters/syllables
  - Type-token ratio, function word distributions
  - Syntactic pattern features, content words, character n-grams

10.7. Explain what plagiarism detection is

- Task: given two texts determine if they are from the same author

10.8. Give a basic idea of how a supervised plagiarism detection system would work

- Machine learning approaches with semantic similarity features
  - Consider pairs of sentences/paragraphs for semantic similarity
  - Additional features that detect "literal" copying of text
- Issue: very hard to distinguish correctly cited parts of text from plagiarised ones

10.9. Describe the difference between single- and multi-document summarization

- Single document summarization improves search:
  - Decide if you really want to read a whole document
  - Acquire the most important information from a document
  - Acquire parts of a document most relevant to a query
- Multi document summarization helps in compiling and presenting:
  - Efficient browsing and searching through big document collections
  - Clustering similar documents and getting a summary of similarities
  - Removing redundancies stemming from different documents on the same topic

10.10. Explain the difference between an indicative and an informative summary

- An indicative summary is intended to help the user decide to read or not to read the whole text while informative summary tries to minimize the difference between the summary and the original. E.g. news headlines are indicative summaries.

10.11. Describe the difference between an extractive and an abstractive summary

- An extractive summary retains only an (unchanged!) subset of the original text.
- An abstractive summary is creating by understanding the information contained in the text, and generating a new shorter natural language text
  - Involves natural language generation (NLG)

- Abstractive summaries are more complex to generate automatically

10.12. Explain the difference between generic and query-oriented summaries

- Generic summaries summarize the overall content of the document
- Query oriented summaries focus the summary on particular parts of the document relevant for a user query

10.13. Explain the basic methods for determining sentence relevance

- Extract sentences containing important terms. Importance of terms defined using:
  - Probability of a (content) word in the collection
  - TF-IDF weights
- Proximity of important terms
  - Terms close to important terms are considered more important. We could use PageRank for this.
- Importance may depend on the number of repetitions, synonyms, coreferents, and related terms that together form a lexical chain (e.g., Rome → Italy → city → inhabitant)
- Sentences cued by indicator phrases (often rhetorical) that indicate important content E.g., "We conclude that. . . "
- Sentences cued by their location E.g., the first sentence of a paragraph is likely to be important

10.14. Explain in brief methods for determining sentence relevance based on supervised machine learning

- Idea: learn a model that determines importance of sentences using a training corpus of full texts and their summaries
- Models: Naive Bayes, Decision trees, SVM, HMM, CRF …
- Features:
  - Sentence length, presence of indicator phrases
  - Sentence position (first/medium/final)
  - Highly weighted content words
  - Containment of (important) named entities
  - Containment of specific topic words
- Topic words: a very useful feature for ML models (Decide whether a word is a topic word (a binary decision)):
  - t : a word from the input
  - T : cluster of articles on given topic
  - NT : articles not on the topic from T

    Words that have similar probability in T and NT are not topic words

    $H1 :\ P(w|T)\ =\ P(w|NT)\ (w\ is\ not\ descriptive\ of\ the\ topic)$

    $H2 :\ P(w|T)\ >\ P(w|NT)\ (w\ is\ descriptive\ of\ the\ topic)$

    The decision can be made using a log-likelihood ratio statistical test

10.15. Explain how to optimally select which sentences to put into the summary

- Often used after determining sentence/term importance
- Define:
  - $Rel(i)$ - relevance of sentence i for the summary
  - $Red(i,j)$ - redundancy between sentences i and j
  - $l(i)$ - length of sentence i
- Goal is to maximize relevance while minimizing redundancy and making the summary short
- The optimal summary S of length no more than K sentences can be found by solving the following optimization problem:
  - $S = \arg\max_{S} \left(\sum_{t_i \in S} Rel(i) - \sum_{t_i, t_j \in S} Red(i,j) \right)$
  - Such that: $\sum_{t_i \in S} l(i,j) \leq K$
- Optimizes all three criteria simultaneously. Can be tackled with greedy search, dynamic programming (DP), integer linear programming (ILP).

10.16. Describe issues that are typical of multi-document summarization and give a basic idea of how to solve them

- Redundancy
- Solution:
  - clustering of sentences in the input:
    - Clusters of similar sentences represent a theme in the output
    - Clusters with many sentences are more important
    - Select one sentence per important cluster

  Sentences are clustered by measuring semantic similarity between sentences

10.17. List the four possible methods of evaluating automated summarization

- Fully automatic method
  - Compares the summary to the original text using automated semantic similarity measures
- Responsiveness method
  - Humans numerically rate the quality of the summary
  - No reference summaries are needed
  - Problems with task subjectivity!
- Pyramid method
  - Humans manually compare generated and reference summaries using a well defined methodology
- ROUGE measures
  - Automatic comparison of system-generated and reference summaries
  - High correlation with manual evaluation

- Variants take into account
    - N-gram co-occurrence
    - Weighted longest common subsequence
    - Skip n-gram co-occurrence

10.18. Describe the task of (lexical) text simplification

- Basic idea: replace long and informationally complex words with their simpler and shorter synonyms
- Identifying synonyms using resources like WordNet. Determining if a synonym is simpler than the original word by looking at its frequency - less frequent words are informationally more complex, more common words are deemed simpler

10.19. Explain in short how lexical text simplification is typically done

- Supervised learning simplification rules from sentence-aligned regular corpora (Wikipedia) and manually-simplified corpora (Simple Wikipedia)
- Unsupervised – learning simplification rules from document-aligned corpora or from standard-language corpora only

10.20. Describe the idea behind content-based text simplification

- Idea: Remove content not relevant for the topic. Descriptive sentence parts, those not relating to event mentions, may be eliminated.
- Event-centered simplification
    - Extract event mentions (anchors and arguments)
    - Remove words not belonging to any event
    - Transform each event mention into a separate sentence
    - Apply rules for retaining grammaticality

10.21. Explain in short how text simplification can be evaluated

- Automated readability evaluation: Kincaid-Flesch Grade Level, SMOG Index, average sentence length, average word length, . . .
- Manual evaluation of (1) simplicity, (2) grammaticality, and (3) meaning preservation

11. Question Answering

11.1. Describe the difference between domain-specific and open-domain QA

- Domain-specific QA
    - Provides answers from questions from a restricted domain Restricted domain: circumscribed, complex, practical Canned QA: questions matched against a set of predetermined questions for which answers are known (FAQ) (aka FAQ-retrieval)
- Open-domain QA (domain-independent QA)
    - Provide answers to questions from any domain by employing NLP techniques on large document collections Challenge: broad range of questions that need to be handled

11.2. Explain the QA inherent problems

- Questions by answer type:
  - Factual questions:
    - Who was the first American in space?
    - When did the French revolution take place?
  - Opinion questions
  - Summary questions
- Kinds of questions:
  - Yes/No
  - “Wh” questions (Who? What? Why? When? How?)

  Indirect requests (I would like you to list. . .)

  Commands (Name all the presidents. . .)
- Hard questions:
  - Why and how questions: require understanding causality or instrumental relations (typically expressed as clauses or separate sentences). How did Socrates die? Why is global warming happening?
  - What questions are often hard because of an insufficiently narrowed answer type
  - What happened? vs. What did they see? vs. What did they do?
- Types of answers
  - Answer form:
    - Long vs. short E.g., justification requires a long answer
    - List vs. narrative
  - Answer generation:
    - Extraction (c/p of snippets containing the answer)
    - Generation

    Example:

    Q: Who wrote Macbeth?

    A (extraction):

    When writing the play Macbeth, Shakespeare created an atmosphere around the characters and the overall setting of the play, with his use of massive amounts of imagery in Macbeth...

    A (generation):

    Macbeth was written by English poet and playwright William Shakespeare.

11.3. Briefly explain the basic idea behind implementing factoid QA

- QA systems capable of processing factual questions
- Extract answer as a text snippet. The answer is typically found verbatim in text.
- Questions are classified by the expected answer type (person, organization, quantity, living entity, ...). The answer must match the expected NE/lexico-semantic class

- NLP techniques: chunking, paraphrase recognition, NER, lexico-semantic classification (WordNet)
- Example:
  Q: Who is the author of the book “Rare Earth”?
  A: Rare Earth by Peter D. Ward and Donald Brownlee. Review by Don DeYoung. This book has become a best seller. The authors, professors at the University of Washington, Seattle, have dared to suggest...

11.4. Briefly explain what resources we could use to perform simple reasoning for QA

- Wordnet
- derivational similarity (poisonous is derived from poison)
- distributional semantic similarity
- parse structure of sentence

11.5. Describe what is involved in the question processing QA step

- Determine the expected answer type and where to search for answers
- Assumption: the answer is in a text snippet that:
  - contains the most representative questions concepts
  - includes a concept of the same category as the expected answer type
- For open-domain QA, relevant paragraphs are retrieved using a keyword-based query Candidate keywords:
  - quoted expressions
  - named entities
  - complex nominals
  - all nouns + adjectival modifiers + main verbs
- Keywords are added/dropped to maximize the plausibility of the answer

11.6. Describe briefly the goal of document processing QA step

- Splitting of documents into paragraphs using topic segmentation (unless the engine allows passage retrieval)
- We end up with a set of candidate paragraphs
- NLP processing:
  - POS tagging,
  - syntax,
  - chunking,
  - NER
  - off-line: on the complete collection (before or after document retrieval
  - on-line: on the retrieved paragraphs (after retrieval)
- If NLP processing is done before retrieval, it can be used to re-rank the retrieved documents

11.7. Explain the basics of the supervised learning approach to the answer extraction QA step

- Candidate answer contains a string whose semantic type matches that of the expected answer
- A classifier trained on an “answer window features” (Moldovan et al, 2002)

- Word overlap between the question and the string of the expected answer type
- Word overlap between the question and the sentence containing the string of the expected answer type
- Is the string of the expected answer type followed by a punctuation
- Position of the text span among all spans from the hit list

12. Sentiment Analysis

12.1. Define sentiment analysis (or opinion mining)

- Sentiment analysis or opinion mining is the computational study of people's opinions, appraisals, attitudes and emotions
    - Toward entities, individuals, issues, events, topics and their attributes (aspects)

12.2. Explain what a sentiment lexicon is

- Sentiment lexicon is a collection of "Sentiment clues"
    - Sentiment clues (opinion words, sentiment-bearing words) – words and phrases used to express some desired or undesired state
        - Some positive clues: good, amazing, beautiful
        - Some negative words: bad, awful, terrible, poor

12.3. Describe the basic idea behind dictionary-based approaches to generating sentiment lexicons

- Idea is to use a small seed sentiment lexicon and exploit semantic links between words in a large general lexicon. (Bootstrapping)
- Shortcoming: Unable to find sentiment clues with context- or domain-specific orientations

12.4. Describe the basic idea behind corpus-based approaches to generating sentiment lexicons

- Idea is to use syntactic relations and co-occurrences found in corpus to construct a graph.
- This makes the resulting sentiment lexicon domain adjusted.

12.5. Describe the limitations of using sentiment lexicons in sentiment analysis

- Sentiment lexicons perform poorly with informal texts because of poor generalization.
- Also they are limited only to languages for which large general lexicons have been built.

12.6. Describe how supervised sentiment classification works

- Supervised sentiment classification works by classifying an opinionated portion of text typically into one of three categories: positive, negative, neutral.
- The most common algorithms used are SVM and Naive Bayes.
- Classification is usually based on features like: bag-of-words (with TF-IDF weighting scheme), parts of speech, opinion clues and phrases, negations, syntactic dependencies.

12.7. Describe the basic idea behind unsupervised sentiment classification approaches

- Basic idea is to extract the phrases (e.g., matching POS-patterns) from text and score them in relation to positive and negative seed words. After sentiment orientations for each phrase is computed, document is classified by averaging them out.

12.8. Explain how a two-step sentiment classification for sentences is performed

- By first performing subjectivity classification on sentences and then Polarity classification on the subjective ones.
- Subjectivity classification - Sentence is classified as subjective or objective
- Polarity classification - Subjective sentences are classified as positive, negative, or neutral

12.9. Describe aspect-oriented sentiment analysis and its advantages over regular sentiment analysis

- Aspect-oriented sentiment analysis is able to analyze different aspects of the same entity and is therefore more domain adjusted.

12.10. Explain in brief three steps typically performed for aspect-oriented sentiment analysis

- Acquisition of domain-specific sentiment lexicon
  - Lists of positive and negative sentiment clues
  - Additionally: A lexicon of domain-specific aspects
- Classification of sentiment for each aspect
  - Model for assigning sentiment clues to aspects
- (Optional) Using aspect-bound sentiment for classification
  - Aspect-oriented sentiment as features for review sentiment classification or regression

12.11. Explain what sarcasm is and one possible approach (discussed in class) how to detect certain forms of sarcasm

- Sarcasm is a sharp, bitter, or cutting expression or remark; a bitter gibe or taunt
- We can detect sarcasm as contrast between negative situation and positive sentiment.