Information Retrieval Exercises

Assignment 4:

Synonym Expansion with Lucene

Mario Sänger (saengema@informatik.hu-berlin.de)

Synonym Expansion

- Idea: When a user searches a term K, implicitly search for all synonyms of K
  - S AND T=>(S OR S’ OR S” OR ..) AND (T OR T’ OR T” OR ...)
- Popular method
- Usually increases recall and decreases precision
- Requires a high quality synonym lexicon
- Can be extended to also include hyponyms (‘banana’ is a hyponym to ‘fruits’)

WordNet

- Lexical database with semantic relationships
- Maintained since 1985
- Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets)
- ~66.000 words, ~180.000 Synsets

- Contains different relationship types: hypernomy, hyponomy, causation, antonomy, holonym, meronym ...

Example Synsets from WordNet

- **[well]**: [considerably] [intimately] [easily] [comfortably] [wellspring] [substantially] [advantageously] [good] [swell] [fountainhead]
- **[good]**: [commodity] [expert] [sound] [respectable] [secure] [estimable] [effective] [honest] [serious] [ripe] [near] [unspoiled] [dear] [just] [salutary] [goodness] [proficient] [skilful] [adept] [thoroughly] [soundly] [unspoilt] [dependable] [right] [upright] [beneficial] [safe] [well] [honorable] [full] [practiced] [skillful]
- **[better]**: [expert] [meliorate] [sound] [respectable] [best] [secure] [good] [estimable] [wagerer] [effective] [honest] [serious] [ripe] [easily] [near] [unspoiled] [dear] [just] [salutary] [proficient] [skilful] [adept] [break] [bettor] [amend] [considerably] [intimately] [unspoilt] [dependable] [comfortably] [right] [upright] [ameliorate] [improve] [beneficial] [safe] [well] [punter] [substantially] [advantageously] [honorable] [full] [practiced] [skillful]

WordNet Online

- You can search synsets directly at WordNet:
  - http://wordnetweb.princeton.edu/perl/webwn

Relationship Types

- Antonyms are words with opposite meanings
  - bad is an antonym of good
- Hyponyms are specific instances of a category
  - red is a hyponym of color
- Hypernyms describe categories of instances
  - color is a hypernym of red
- Holonyms define a relationship between terms (one is part of the other):
  - tree is a holonym of trunk
- Meronyms are the opposite of holonyms:
  - trunk is a meronym of tree

Task

- Implement synonym expansion within Lucene (v7.3.1) for the IMDB movie plots
- You can reuse your existing code from assignment 3
  - Using word tokenization and stop word removal, no stemming
- Use WordNet as lexicon
  - Current release: WordNet 3.1
- For simplicity, we will only consider Boolean (AND, OR, NOT) term search
- No phrase or proximity search any more

Query Expansion in Lucene

- Option 1: At indexing time
  - Add all expansions to all terms of a document d when indexing d
- Option 2: At search time
  - When searching a keyword K, rewrite query in disjunction of all expansions of K
  - Original Query: plot:Berlin AND plot:wall AND type:television
  - Extended Query: plot:berlin AND (plot:bulwark OR plot:fence OR plot:palisade OR plot:paries OR plot:rampart OR plot:surround OR plot:wall) AND (type:telecasting OR type:television OR type:telly OR type:tv OR type:video)
- Note: If K is part of more than one synset, use all
  - No disambiguation

Getting Started

- Download WordNet 3.1 files at
  - http://wordnetcode.princeton.edu/wn3.1.dict.tar.gz
- Extract noun, verb, adj, adv files:
  - data.[noun, verb, adj, adv] (synsets)
  - [noun, verb, adj, adv].exc (base forms)
- Parse synsets from these plain files using syntax:
  - https://wordnet.princeton.edu/documentation/wndb5wn

Date File Format (synsets)

- Each data file begins with a copyright notice - skip this!
- Each synset is encoded in one line:
  - *synset_offset lex_filenum ss_type w_cnt word lex_id [word lex_id...] p_cnt [ptr...] [frames...] | gloss*
  - *w_cnt*: Two digit hexadecimal integer indicating the number of words.

- Example line (synset):
  00007846 03 n **06 person** 0 **individual** 0 **someone** 0 **somebody** 0 **mortal** 0 **soul** 0 421 @ 00004475 n 0000 @ 00007347 n 0000 #m 07958392 n 0000 + 01562007 a 0501 + %p ...

Exception List File Format

- The first field of each line is an inflected form, followed by a space separated list of one or more base forms of the word

- Examples:
  - better good well
  - bigger big

- Meaning: all synsets of good and well apply to better (but not the reverse!!)

Complications I

- Use only single-token synonyms
  - Ignore all synonyms with more than one token
  - These are formatted by a "_" in the name (e.g., house_of_cards)

- Special adjective syntax
  - Remove (p), (a) and (ip) from adjectives (e.g. galore(ip))
  - https://wordnet.princeton.edu/documentation/wninput5wn

Complications II

- Merge synsets of words appearing in the verb, nouns, adj, adv files
  - For example: reason (noun) and reason (verb)
- Consider a synset as set
  - Example: Synset of cause = {reason,grounds}
  - Create the following synonym relations: cause-reason, cause-grounds, reason-grounds and all reverse relations reason-cause, grounds-cause, grounds-reason
- BUT do not apply this rule transitively
  - Example: cause = {grounds} and grounds={earth} should not create cause-earth!
  - Syn-relationships in WordNet do not form an equivalence class!

Complications III

- The exception lists are not symmetric
  - The inflected form is merged with all synsets of its base forms but not the reverse
- An exception given in adj.exc only adds the synsets defined in the data.adj file. An exception in noun.exc only adds the synsets defined in the data.noun file.
  - So you have to keep the synsets in noun, adj, adv, verb separated for the exception lists
- Example: Given an exception in adj.exc: better good well
  - syns (better):=$syns_{adj}$(better)∪ $syns_{adj}$(good) ∪ $syns_{adj}$(well) ∪ good ∪ well
  - But not syns(well) := $syns_{adj}$(better) ∪ ...
  - And not syns(better) := $syns_{noun}$ (better) ∪ ... $syns_{noun}$(well)

Complications IV

- The exception files define base and inflected forms for irregular words
  - WordNet applies lemmatization for regular words based on rules like big, bigger, biggest
  - https://wordnet.princeton.edu/documentation/morphy7wn
  - But you can skip this!

- Some true results for reference
  - Only sysnets: 60993 words with 153394 synonyms
  - Synsets & exception lists: 66126 words with 176476 synonyms

BooleanSearchWordnet.java

- public void buildSynsets(Path wordnetDir)
  - Used to parse the wordnet files and build the synonym index
- public void buildIndices(Path plotFile)
  - Used to parse the file and build the Lucene index
- public Set<String> booleanQuery(String queryString)
  - Parses the query string and returns the title lines of any entries in the plotFile matching the query
- public void close()
  - Can be used to close used resources (e.g. Lucene index, thread pool, etc.)

Test your program

- We provide you with:
  - queries_wordnet.txt: file containing exemplary queries
  - results_wordnet.txt: file containing the expected results of running these queries
  - a main method for testing your code (which expects as parameters the corpus file, the queries file and the results file)

- You can check your synonym expansion for plausibility on the WordNet website:
  - http://wordnetweb.princeton.edu/perl/webwn

Submission

- **Group 1: Friday, 29.06., 23:59 (midnight)**
- **Group 2: Sunday, 01.07., 23:59 (midnight)**

- Submit a ZIP archive named *ass4_<group-name>.zip*
  - Java source files of your solution
  - Compiled and executable BooleanQueryWordnet.jar

- Upload archive to the HU-BOX:
  https://box.hu-berlin.de/u/d/32fe78ed297444c2a9bb/

Submission requirements

- Test your jar before submitting by running the examples queries on one of the gruenau hosts
  - java -jar BooleanQueryWordnet.jar <plot list file> **<wordnetDir>** <queries file> <results file>
  - You might have to increase the JVM's heap size (e.g., -Xmx8g)
  - Your jar must run and answer all test queries correctly!

- Your program has to correctly answer all example queries correctly to pass the assignment!

Solution presentations

- The presentation of the solutions will be given on 09.07. resp. 11.07.
- You are be able to pick when and what you'd like to present (first-come-first-served):
  - Group 1 (Mo): https://dudle.inf.tu-dresden.de/ire_ass4_mo/
  - Group 2 (We): https://dudle.inf.tu-dresden.de/ire_ass4_we/
- Presentation of the following aspects:
  - Lucene WordNet Indexer
  - Lucene Query Expansion

Competition

- Search as fast as possible
- Stay under 50 GB memory usage
- We will call the program using our evaluation tool:
  - We will use different queries and -Xmx50g parameter

- Evaluation will be twofolded again:
  - The total query time
  - The total time for building the index

Submission checklist

1. Did not change or remove any code from BooleanQueryWordnet.java
2. Did not alter the functions‘ signatures (types of parameters, return values)
3. Only use the default constructor and don‘t change its parameters
4. Did not change the class or package name
5. Named your jar BooleanQueryWordnet.jar
6. Tested your jar on gruenau hosts by running
   java -jar BooleanQueryWordnet.jar plot.list wordNetDir queries.txt results.txt
   - You might have to increase Java heap space (e.g. -Xmx6g)
7. Ascertained that the queries in queries_wordnet.txt were answered correctly
8. Make sure to upload a zip file named by your group name

Timetable / Next steps

- Assignment 4 submission deadline:
  - **Group 1: Friday, 29.06., 23:59 (midnight)**
  - **Group 2: Sunday, 01.07., 23:59 (midnight)**

- Presentations of the solutions for assignment 3
  - **Group 1: Monday, 25.06.**
  - **Group 2: Wednesday, 27.06**