Information Retrieval Exercises

Assignment 4:

Synonym Expansion with Lucene

Samuele Garda (gardasam@informatik.hu-berlin.de)

Query Expansion

- Augmenting a given query to improve retrieval performance

- Synonym Expansion:
  - search for term K = implicit search for all synonyms of K as well:
    - S AND T=>(S OR S’ OR S’’ OR ..) AND (T OR T’ OR T’’ OR ...)

- Usually increases recall and decreases precision

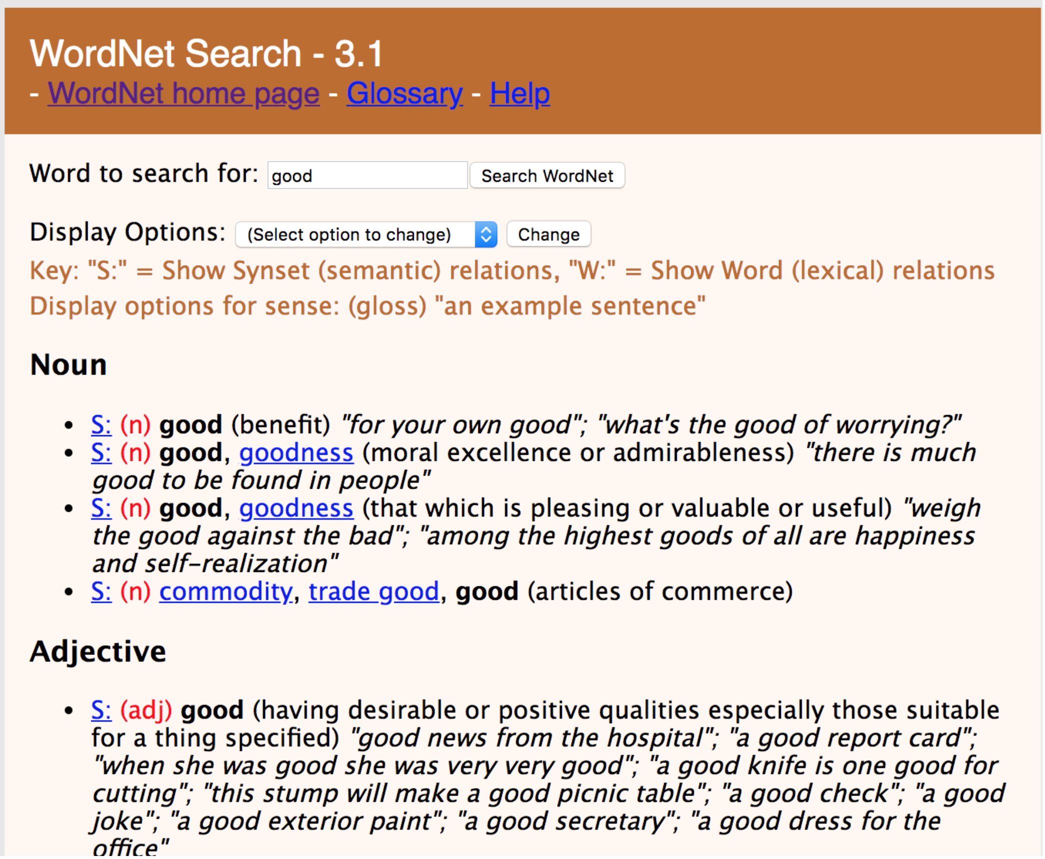
- Requires a high quality synonym lexicon

WordNet

- Lexical database, maintained since 1985:
  - https://wordnet.princeton.edu/
- Nouns, verbs, adjectives and adverbs are grouped into sets of “cognitive” synonyms (synsets):
  - ~66.000 words, ~180.000 Synsets
- Contains different relationship types:
  - Antonomy : “wet” ↔ “dry”
  - hypernomy ↔ hyponomy : “furniture” ↔ “bed”
  - holonym ↔ meronym: “finger” ↔ “hand”

WordNet Online

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



WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: good Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **good** (benefit) *"for your own good"; "what's the good of worrying?"*
- S: (n) **good**, goodness (moral excellence or admirableness) *"there is much good to be found in people"*
- S: (n) **good**, goodness (that which is pleasing or valuable or useful) *"weigh the good against the bad"; "among the highest goods of all are happiness and self-realization"*
- S: (n) commodity, trade good, **good** (articles of commerce)

Adjective

- S: (adj) **good** (having desirable or positive qualities especially those suitable for a thing specified) *"good news from the hospital"; "a good report card"; "when she was good she was very very good"; "a good knife is one good for cutting"; "this stump will make a good picnic table"; "a good check"; "a good joke"; "a good exterior paint"; "a good secretary"; "a good dress for the office"*

Task

- Implement synonym expansion within Lucene (v8.8.2)

- You can reuse your existing code
  - Using **word tokenization** and **stop word removal**, **no stemming**

- Use WordNet as lexicon (current version: 3.1)

- Use IMDB movie corpus (“plot.list” filr)

Task

- For simplicity, we will only consider Boolean query (AND, OR, NOT) and term search
- No phrase or proximity search any more
- Note: If K is part of more than one synset, use all
  - i.e. no disambiguation

Query Expansion in Lucene

- 1) At indexing time
  - Add all expansions to all terms of a document D when indexing

- 2) At search time
  - When searching a keyword K, rewrite query in disjunction of all expansions of K, e.g.:
  - plot:Berlin AND plot:wall AND type:television

    →
  - 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)

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

Data File Format: synsets (.data)

- 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 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 ...

Data File Format: base forms (.exc)

- 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, e.g.:
  - better good well
  - bigger big

- The exception lists are not symmetric
  - The inflected form is merged with all synsets of its base forms but not the reverse
  - Meaning: all synsets of good and well apply to better, **but not the inverse**!

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)
  - See “Special Adjective Syntax” section:
    - https://wordnet.princeton.edu/documentation/wninput5wn

Complications II

- Consider a synset as set
  - Example: cause = {reason,grounds}
  - Synonym relations: cause-reason, cause-grounds, reason-grounds
  - reverse relations reason-cause, grounds-cause, grounds-reason
- Do **NOT** apply this rule **transitively**
  - Synet relationships in WordNet **do not form an equivalence class**
  - they do not have the transitivity property
    - cause ~ ground ^ ground ~ earth ⇏ cause ~ earth

Complications III

- An exception given in XXXX.exc only adds the synsets defined in the data.XXXX file.

- So you have to keep the synsets in noun, adj, adv, verb separated for the exception lists

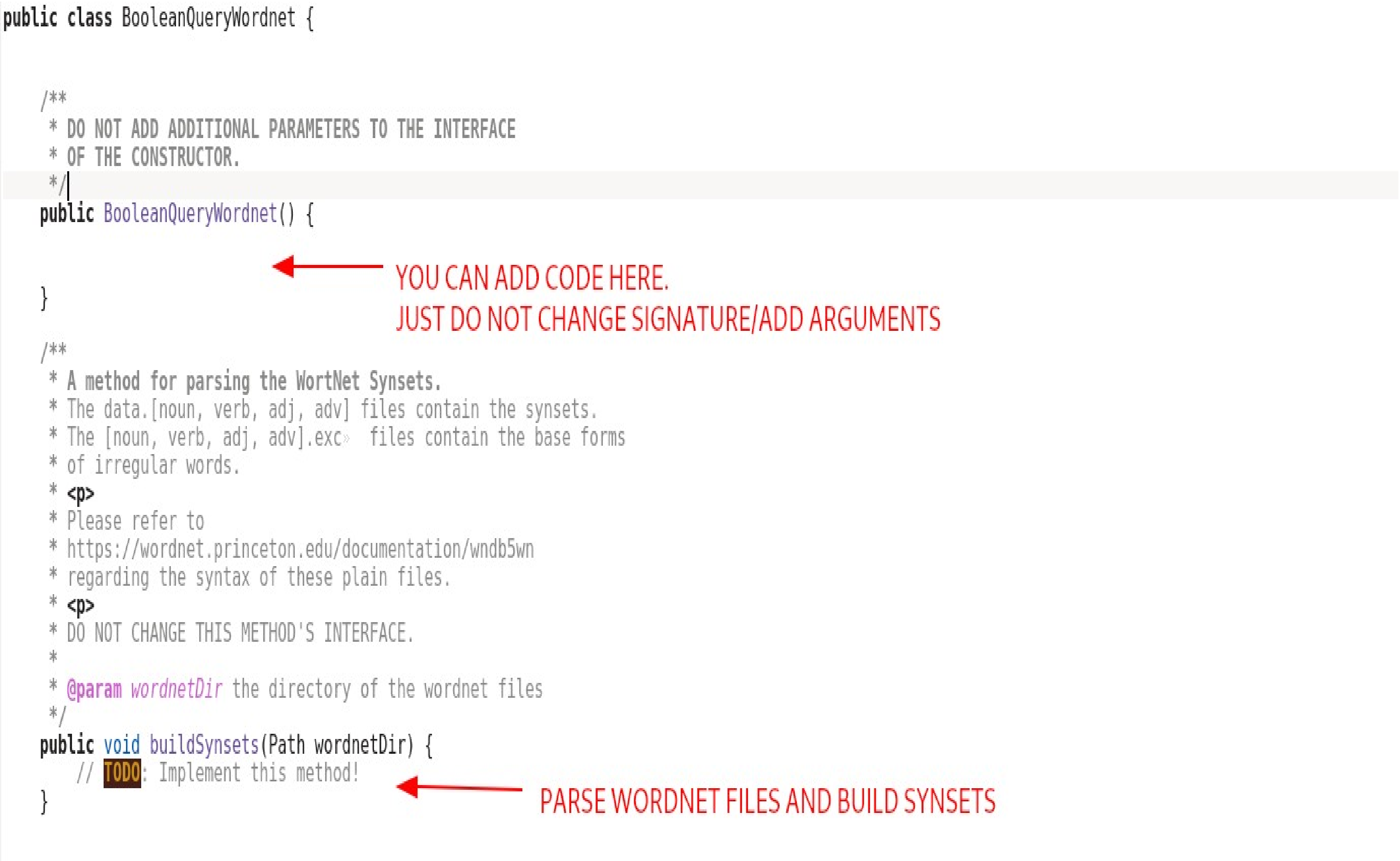
- Given an exception in adj.exc, e.g. “better good well”:
  - $\text{syns (better)} := \text{syns}_{\text{adj}}(\text{better}) \cup \text{syns}_{\text{adj}}(\text{good}) \cup \text{syns}_{\text{adj}}(\text{well}) \cup \text{good} \cup \text{well}$
  - $\text{syns(well)} :\neq \text{syns}_{\text{adj}}(\text{better}) \cup \dots$
  - $\text{syns(better)} :\neq \text{syns}_{\text{noun}}\ (\text{better}) \cup \dots\ \text{syns}_{\text{noun}}(\text{well})$

For Your Information

- 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 this is not relevant for the assignment!

The code

The code

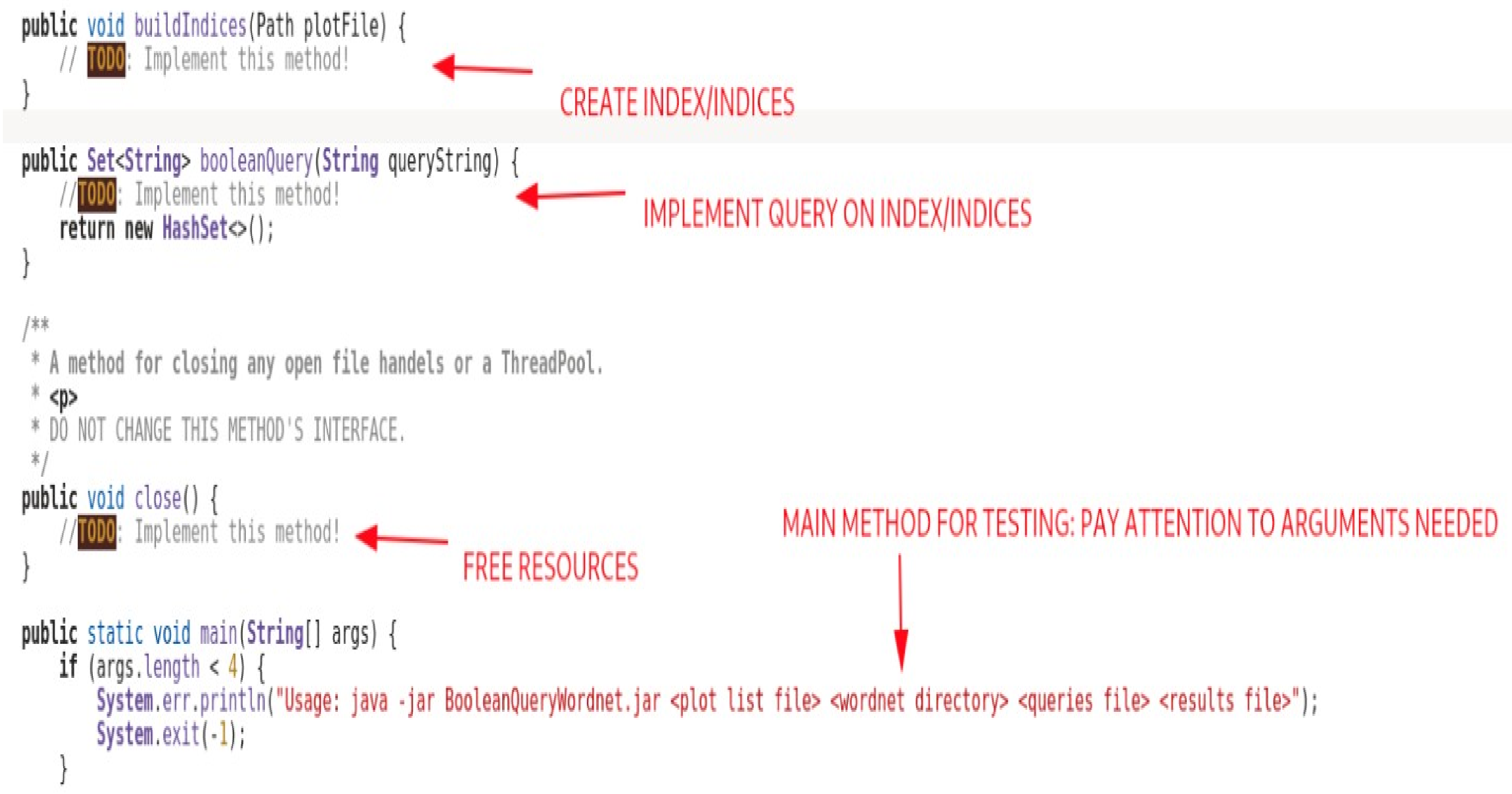


```
public class BooleanQueryWordnet {


    /**
     * DO NOT ADD ADDITIONAL PARAMETERS TO THE INTERFACE
     * OF THE CONSTRUCTOR.
     */
    public BooleanQueryWordnet() {


    }

    /**
     * A method for parsing the WortNet Synsets.
     * The data.[noun, verb, adj, adv] files contain the synsets.
     * The [noun, verb, adj, adv].exc»  files contain the base forms
     * of irregular words.
     * <p>
     * Please refer to
     * https://wordnet.princeton.edu/documentation/wndb5wn
     * regarding the syntax of these plain files.
     * <p>
     * DO NOT CHANGE THIS METHOD'S INTERFACE.
     *
     * @param wordnetDir the directory of the wordnet files
     */
    public void buildSynsets(Path wordnetDir) {
        // TODO: Implement this method!
    }
```

YOU CAN ADD CODE HERE.
JUST DO NOT CHANGE SIGNATURE/ADD ARGUMENTS

PARSE WORDNET FILES AND BUILD SYNSETS

The code



```
public void buildIndices(Path plotFile) {
    // TODO: Implement this method!
}

public Set<String> booleanQuery(String queryString) {
    //TODO: Implement this method!
    return new HashSet<>();
}

/**
 * A method for closing any open file handels or a ThreadPool.
 * <p>
 * DO NOT CHANGE THIS METHOD'S INTERFACE.
 */
public void close() {
    //TODO: Implement this method!
}

public static void main(String[] args) {
    if (args.length < 4) {
        System.err.println("Usage: java -jar BooleanQueryWordnet.jar <plot list file> <wordnet directory> <queries file> <results file>");
        System.exit(-1);
    }
```

CREATE INDEX/INDICES

IMPLEMENT QUERY ON INDEX/INDICES

FREE RESOURCES

MAIN METHOD FOR TESTING: PAY ATTENTION TO ARGUMENTS NEEDED

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

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

Submission requirements

- Test your jar before submitting by running the examples queries on gruenau
  - 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 program has to correctly answer all example queries correctly to pass the assignment!**

Submission

- Make sure that you…
  - ... did not change or remove any code from BooleanQueryWordnet.java
  - ... did not alter the functions‘ signatures (types of parameters, return values)
  - ... only use the default constructor and don‘t change its parameters
  - ... did not change the class or package name
  - ... named your jar BooleanQueryWordnet.jar

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

Timetable / Next steps

- Assignment 4 submission deadline:
  - **Group 1: Tuesday, 29.06., 23:59 (midnight)**
  - **Group 2: Wednesday, 30.06., 23:59 (midnight)**
- QA session in between
- Presentations of the solutions for assignment 4
  - **Group 1: Monday, 06.07.**
  - **Group 2: Wednesday, 07.07**
- Presentation of the following aspects:
  - Lucene WordNet Indexer
  - Lucene Query Expansion

Questions?