As we conclude, it is clear that the field of NLU is continuously evolving. The interplay between symbolic and subsymbolic techniques will likely drive future advancements, pushing the boundaries of what machines can comprehend and achieve. Our exploration has underscored the importance of ongoing research and innovation in developing systems that not only process language but also understand and reason with it in ways that mirror human intelligence. The future of NLU holds promise for more intuitive and intelligent interactions between humans and machines. By addressing current challenges and leveraging the strengths of both symbolic and subsymbolic approaches, we can look forward to a new era of AI that truly understands the nuances of human language. This book serves as a testament to the progress made thus far and a guidepost for the exciting developments yet to come.

6.1 Learning Resources

6.1.1 Assignment

In this group assignment, you will build an opinion search engine. Given a specific topic of your choice (e.g., cryptocurrencies), your system should enable users to find relevant opinions about any instance of such topic (e.g., bitcoin) and perform sentiment analysis on the results (e.g., opinions about bitcoin are 70% positive and 30% negative). Once you have chosen a topic, make sure that a) you can find enough data about it (e.g., some topics may be too niche to the point that you would only find a few hundreds data about it) and b) the opinions you get are balanced (e.g., if the topic you chose only has negative opinions associated with it, then it is probably not a good topic). For ideas about interesting topics, you can check our project page at https://sentic.net/projects. You are pretty much free to use anything you want in terms of available tools/libraries. However, your system cannot be just a mashup of existing services. Your final score will depend not only on how you developed your system but also on its novelty and your creativity: in other words, to get a high score you do not only need to implement a system that works, but also a system that is useful and user-friendly. The main tasks of the assignment are crawling (20 points), indexing (40 points) and classification (40 points). A minimum of 60 points is required to pass the assignment. Consult with your course coordinator regarding the procedures and requirements for submitting your assignment.

A. Crawling (20 points)

Crawl text data from any sources which you are interested in and permitted to access, e.g., 𝕏 API or Reddit API. The crawled corpus should have at least 10,000 records and at least 100,000 words. It is fine to use available datasets for training (e.g., popular sentiment benchmarks), but you still have to at least crawl and label data for testing.

Also, make sure your dataset does not contain duplicates and try your best to make it balanced (e.g., equal number of positive and negative entries). Before crawling any data, carefully consider the questions in this material, e.g., check whether the data have enough details to answer the questions. You can use any third party libraries for the crawling task, e.g.:

- Jsoup: https://jsoup.org
- Twitter4j: https://twitter4j.org
- Facebook marketing: https://developers.facebook.com/docs/marketing-apis
- Instagram: https://instagram.com/developer
- Amazon: https://github.com/ivanpgs/amazon-crawler
- Tinder: https://gist.github.com/rtt/10403467
- Tik Tok: https://developers.tiktok.com

Question 1: Explain and provide the following:

1) How you crawled the corpus (e.g., source, keywords, API, library) and stored it
2) What kind of information users might like to retrieve from your crawled corpus (i.e., applications), with sample queries
3) The numbers of records, words, and types (i.e., unique words) in the corpus

B. Indexing (40 points)

Indexing: You can do this from scratch or use a combination of available tools, e.g., Solr+Lucene+Jetty. Solr runs as a standalone full-text search server within a servlet container such as Jetty and it uses the Lucene search library at its core for text indexing and search. Solr has REST-like HTTP/XML and JSON APIs that make it easy to use from any programming language. Useful documentations include:

- Solr project: https://solr.apache.org
- Solr wiki: https://wiki.apache.org/solr/FrontPage
- Lucene tutorial: https://lucene.apache.org/core/quickstart.html
- Solr with Jetty: https://wiki.apache.org/solr/SolrJetty
- Jetty tutorial: https://jetty.org

You can also choose other inverted-index text search engine open projects, e.g., Sphinx, Nutch, and Lemur. However, you should not simply adopt SQL-based solutions for text search (for example, you cannot solve text search simply using Microsoft Sqlserver or MySql).

Querying: You need to provide a simple but friendly user interface (UI) for querying. It could be either a web-based or mobile app based UI. You could use JSP in Java or Django in Python to develop your UI website. Since Solr provides REST-like APIs to access indexes, one extra JSon or RESTFul library would be enough. Otherwise, you may use any third party library.

The UI must be kept simple: a sophisticated UI is not necessary nor encouraged. Detailed information besides text is allowed to be shown for the query results, e.g., product images on Amazon, ratings on Amazon, and pictures on Instagram. The details should be designed to solve specific problems.

Question 2: Perform the following tasks:

- Design a simple UI (you can design one from scratch or you can tap on an existing one, e.g., Solr UI) to allow users to access your system in a simple way
- Write five queries, get their results, and measure the speed of the querying

Question 3: Explore some innovations for enhancing the indexing and ranking. Explain why they are important to solve specific problems, illustrated with examples. You can list anything that has helped improving your system from the first version to the last one, plus queries that did not work earlier but now work because of the improvements you made. Possible innovations include (but are not limited to) the following:

- Timeline search (e.g., allow user to search within specific time windows)
- Geo-spatial search (e.g., use map information to refine query results)
- Enhanced search (e.g., add histograms, pie charts, word clouds, etc.)
- Interactive search (e.g., refine search results based on users' relevance feedback)
- Multimodal search (e.g., implement image or video retrieval)
- Multilingual search (e.g., enable information retrieval in multiple languages)
- Multifaceted search (e.g., visualize information according to different categories)

C. Classification (40 points)

Choose two or more subtasks from the NLU suitcase model (Fig. 1.12) to perform information extraction on your crawled data. For example, you could choose *subjectivity detection* and *polarity detection* to first categorize your data as *neutral* versus *opinionated* and then classify the resulting opinionated data as *positive* versus *negative*. Different classification approaches can be applied, including:

- knowledge based, e.g., SenticNet
- rule based, e.g., linguistic patterns
- machine learning based, e.g., deep neural networks
- hybrid (a combination of any of the above)

You can tap into any resource or toolkit you like, as long as you motivate your choices and you are able to critically analyze obtained results. Some possible choices include:

- Weka: https://cs.waikato.ac.nz/ml/weka
- Hadoop: https://hadoop.apache.org
- Pylearn2: https://pylearn2.readthedocs.io/en/latest
- SciKit: https://scikit-learn.org
- NLTK: https://nltk.org
- Theano: https://github.com/Theano
- Keras: https://github.com/fchollet/keras
- Tensorflow: https://github.com/tensorflow/tensorflow
- PyTorch: https://pytorch.org
- Huggingface: https://huggingface.co/
- AllenNLP https://github.com/allenai/allennlp

Question 4: Perform the following tasks:

- Justify your classification approach choice in relation to the state of the art
- Discuss whether you had to preprocess data and why
- Build an evaluation dataset by manually labeling at least 1,000 records with an inter-annotator agreement of at least 80%
- Provide metrics such as precision, recall, and F-measure on such dataset
- Perform a random accuracy test on the rest of the data and discuss results
- Discuss performance metrics, e.g., speed, and scalability of the system

Question 5: Explore some innovations for enhancing classification. If you introduce more than one, perform an ablation study to show the contribution of each innovation. For example, if you perform WSD and NER to enhance sentiment analysis, show the increase in accuracy when adding only WSD, the increase in accuracy when adding only NER, and the increase in accuracy when adding both WSD and NER to your system. Explain why they are important to solve specific problems, illustrated with examples. Possible innovations include (but are not limited to) the following:

- Enhanced classification (add another NLU subtask, e.g., sarcasm detection)
- Fine-grained classification (e.g., perform ABSA)
- Hybrid classification (e.g., apply both symbolic and subsymbolic AI)
- Cognitive classification (e.g., use brain-inspired algorithms)
- Multitask classification (e.g., perform two or more NLU tasks jointly)
- Ensemble classification (e.g., use stacked ensemble)

D. Submission

The submission shall consist of one single PDF file. Add some pictures to your report to make it clearer and easier to read. There is no page limit and no special formatting is required. The file shall contain the following five key items:

1) The names of the group members in the first page
2) Your answers to all the above questions

3) A YouTube link to a video presentation of up to 5 minutes: in the video, introduce your group members and their roles, explain the applications and the impact of your work and highlight, if any, the creative parts of your work (note that you do not have to give all the answers in the video presentation)
4) A Dropbox (or similar, e.g., Google Drive or OneDrive) link to a compressed (e.g., zip) file with crawled text data, queries and their results, evaluation dataset, automatic classification results, and any other data for Questions 3 and 5
5) A Dropbox (or similar, e.g., Google Drive or OneDrive) link to a compressed (e.g., zip) file with all your source codes and libraries, with a README file that explains how to compile and run the source codes

6.1.2 Quiz

1) What is the difference between NLP and NLU?
   A) NLP and NLU are synonymous and can be used interchangeably
   B) NLP focuses on text meaning, NLU on processing and generating text
   C) NLU includes more theoretical tasks, NLP is more focused on applications
   D) NLP handles language interaction, NLU focuses on understanding meaning

2) What does task decomposition typically involve?
   A) Splitting a complex NLP task into simpler, manageable subtasks
   B) Translating text from one language to another
   C) Summarizing long documents into shorter versions
   D) Evaluating the accuracy of an NLP model

3) What is the meaning of panalogy?
   A) Comparing machine learning models using parallel datasets
   B) Using parallel processing to speed up computations
   C) Comparing different concepts by examining their parallel structures
   D) Enhancing natural language processing by using parallel corpora

4) What does symbol grounding refer to in the context of AI?
   A) The process of translating symbols into natural language
   B) The ability to create new symbols for unknown concepts
   C) Relating symbols to real-world objects or concepts to give them meaning
   D) The use of symbols to perform logical operations

5) What is a topology-aware similarity measure?
   A) A method that considers the geometric arrangement of data points
   B) A technique for comparing the colors in images
   C) An approach for evaluating the textual similarity between documents
   D) A measure that assesses the similarity of time series based on frequency