Requirements for an NLP Pipeline

Preprocessing

- tokenization

Splitting a text into its smallest meaningful units (words and punctuation)

- Available:
  - nltk
  - http://polyglot.readthedocs.io/en/latest/Tokenization.html
    - Licence: polyglot_licence. See https://docs.google.com/document/d/1fvFmypstNkxL8-JMWKvLN212cmjYnZLu8LDJYbIfNog/edit#bookmark=id.j8yx3cy4g6uq
- Applied: Yes

Deep learning: too simple to model in a deep network.
Importance: 3

- We have regex based tokenizer, it also works if we cannot use polyglot by licence restrictions

- modelling the vector-space, term weighting

Modelling the literal words / phrases as vectors in the space by weighting them using a method like tf*idf. This is the basic approach for text classification.

- Available: An example can be seen at http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
    - Licence: BSD, allows commercial use
- Applied: yes
    - This is the method with which we build our feature space in the current system

Deep learning: Yes, word embeddings.

- Three sets of vectors are available for turkish
    - Learning word representations for Turkish (Türkçe için kelime temsillerinin öğrenimi) 2014 http://research.sabanciuniv.edu/26655/ ; http://myweb.sabanciuniv.edu/umutsen/research/
    - Named Entity Recognition on Twitter for Turkish using Semi-supervised Learning with Word Embeddings 2016 http://tabilab.cmpe.boun.edu.tr/projects/ttner/
    - Polyglot: Distributed Word Representations for Multilingual NLP 2013 http://polyglot.readthedocs.io/en/latest/Embeddings.html
        - Licence restriction: See https://docs.google.com/document/d/1fvFmypstNkxL8-JMWKvLN212cmjYnZLu8LDJYbIfNog/edit#bookmark=id.j8yx3cy4g6uq
- It is possible to obtain our own embeddings. It is good to have one if the domain of the data is different than those of the available ones (e.g. problem domain is customer emails but the embeddings are obtained mostly from news texts). It would be good that the source texts are well-written, representative and many.

Importance: 3

- spell checking

Checks a word for grammatical errors and fixes it / introduces suggestions if it is problematic.

- Available:
    - Norvig's model: http://norvig.com/spell-correct.html
        - Language adaptable
    - hunspell
        - Command line based: https://github.com/hunspell/hunspell
        - Not tried
    - Aspell
        - Python version: https://github.com/WojciechMula/aspell-python
        - Not very correct
    - Zemberek: https://github.com/ahmetaa/zemberek/tree/master/zemberek-ooo/src/net/zemberek/ooo/spellchecker

        - In Java
- Applied: yes
    - Norvig's model applied for turkish using ~30MB text. It runs very slow, spends ~3 min for a word.
        - Its english version runs properly as a library: https://pypi.python.org/pypi/autocorrect/0.2.0

Deep learning: Yes
- Neural approach not very popular recently: https://scholar.google.com.tr/scholar?as_ylo=2015&hl=en&as_sdt=2005&sciodt=0,5&cites=20314522046311469 21&scipsc=
- Can be obtained as a by-product of normalization

Importance: 3
- The current email data requires more correction than a spellchecker can do, possibly a manual intervention.

- normalization

Corrects the informally written portions in the text (e.g. geliyoruuum->geliyorum; msj->mesaj, emoticons etc). Especially required for social media or speech data.
- Availability:
    - In itü pipeline: http://tools.nlp.itu.edu.tr/Normalization
        - Licence is problematic
    - (probably not published) https://github.com/osmanbaskaya/tr-text-normalization/tree/master/src
        - Haven't tried
- Applied: No

Deep learning: Yes
- An example http://www.aclweb.org/anthology/W15-15#page=20

Importance: 2
- If there is none, the system works less correctly.

Language processing for feature extraction

1)morphological analysis / disambiguation

Splits a word into its morphemes, i.e. its roots and suffixes that are the smallest functional units serving for meaning change or grammatical markers. Required for stemming. Disambiguators also output the POS tags of words.
- Available:

        - In itü pipeline: http://tools.nlp.itu.edu.tr/MorphDisambiguator
            - Licence problems
        - Resources for Turkish morphological processing 2011 http://www.cmpe.boun.edu.tr/~gungort/papers/Resources%20for%20Turkish%20Morphological%20Processing.pdf
            - The tool used to be at http://www.cmpe.boun.edu.tr/~hasim/ (I have the files)
                - Does not run with python3
                - Licence is restricted for commercial use but we can contact
        -

Deep learning: yes
Importance: 2

- Classification is not impossible but less accurate without having the root forms of words
- Required for question answering (QA) / text generation

2) POS tagging

Labelling the words for their types / classes / parts of speech (like noun, verb, adjective). Required for keyword extraction and QA. Very useful for sentiment analysis.

- Availability: see morphological analysis
    - Other options
        - A model for turkish can be developed and used with stanford corenlp tagger: http://nlp.stanford.edu/software/tagger.html
            - Commercial use requires permission, contacted, no answer.
        - Apache opennlp allows training their model. tagged data needed, may be metu-treebank can be converted from conll format to the trainer's format. http://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html#tools.postagger
        - May not be very reliable student projects requiring treebank:
            - https://github.com/onuryilmaz/turkish-pos-tagger
            - https://sirinnes.wordpress.com/tag/turkish-pos-tagger/
        - Turbo parser seems language-independent
            - http://www.cs.cmu.edu/~afm/TurboParser/README
        - metu treebank (http://ii.metu.edu.tr/corpus)
            - Not for commercial purposes.

Deep learning: yes
Importance: 2

- See morphological analysis

3) NER

Identification of named entities (like persons, locations, organizations) in the text.

- Availability:
    - In itü pipeline
        - Licence problem
    - Named Entity Recognition on Twitter for Turkish using Semi-supervised Learning with Word Embeddings 2014 http://tabilab.cmpe.boun.edu.tr/projects/ttner/
        - Tailored for social media texts
    - Polyglot NER: http://polyglot.readthedocs.io/en/latest/NamedEntityRecognition.html
        - Licence: https://docs.google.com/document/d/1fvFmypstNkxL8-JMWKvLN212cmjYnZLu8LDJYbIfNog/edit#bookmark=id.j8yx3cy4g6uq
- Applied: yes, polyglot works well.

Deep learning: yes

- We can try modifying / adapting from (Named Entity Recognition on Twitter for Turkish using Semi-supervised Learning with Word Embeddings)

Importance: 3

- (I can't see an urgency but if we are to detect the sentiment against some named entity, for example, then it is required.)

4) syntactic parse trees, dependency relations

Extracting the parse trees, syntactic roles of the phrases. Useful for semantic processing, relation extraction, sentiment analysis, QA.

- Availability:
    - In itü pipeline
        - Licence problems
    - Syntaxnet: https://research.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html
        - https://github.com/tensorflow/models/tree/master/syntaxnet
    - Berkeley parser: https://github.com/slavpetrov/berkeleyparser
        - Can be trained for tr
        - No licence information
- Applied: Yes. Syntaxnet.

Deep: yes. See syntaxnet.

Importance: 2

- Not very urgent

5) relation extraction

Extracting the relations among entities in text. Useful for semantic processing, sentiment analysis, QA.

- Availability:
    - None for turkish. Some initial works:
        - http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6830184
        - http://www.aclweb.org/anthology/W11-0903
- Applied: No

Deep learning: yes
Importance: 1

Analysis / visualization tasks

- keyword extraction

Obtaining the keywords that are expected to represent the subject of a single text. Useful for analysis before designing a learner or reporting to the end user. Requires a POS tagger.

- Availability:
    - Three tools are available but note for turkish since it lacks a pos tagger.
        - tfidf & textrank: to extract phrases, a pos tagger is needed to capture the structures like (adj noun), (det noun) in both methods
        - rake (rapid automatic keyword extraction): pos tagger not needed but the predefined stopword and phrase patterns in the algorithm have to be modified
- Applied: yes for english. For turkish it gives irrelevant results without syntactic information.

Deep learning: astarı yüzünden pahalı olur diyelim.
Importance: 2

- topic extraction

Extracting the topics in the whole dataset usually with an unsupervised method. Useful for analysis before designing a learner or reporting to the end user. May not require tools more than simple preprocessing and term weighting.

- Availability:
    - See for an example: http://scikit-learn.org/stable/auto_examples/applications/topics_extraction_with_nmf_lda.html

        - Gensim: https://radimrehurek.com/gensim/tutorial.html
    - Applied: yes

Deep learning: Embeddings can be tried.
Importance: 3

Task_name What it does Availability (1) link to the tool if any; 2) applied means we use it in our system Deep learning: possibility of modelling the problem in a deep neural network. Importance scale: 1: low, 2: moderate, 3: high

Polyglot_licence: full gpl licence, requires contact for commercial use
https://sites.google.com/site/rmyeid/projects/polyglot#TOC-Citing-Polyglot