**Engin Ertaş**
**1560176**

# CENG 463: Introduction to NLP
# HW #1- Process Report

## Part I

My class "**stemmer**" inheriting **"nltk.stem.api.StemmerI"** is included in **"hw1.py"** python file. Before programming lines of codes, I had to do some preliminary work. In the begining, I read the documents stated in the homework paper. Then, I tried to prepare a model like in Reference I:

1. I defined a **NFA S= (K, ∑, Δ, s, F)**
2. **K** is the set of states including **18** states (**q0- q16** and start state **s**) which is defined in **"states"** array (hw1.py)
3. **∑** is the alphabet including empty string **ϵ** and **30** possible elements described in the homework paper.
4. **Δ** is the set of transitions, which is defined in **"paths"** array (hw1.py)
5. **s** is the start state linked by empty string **ϵ** to **13** states which are string formations that can be placed to the **end** of a verb. It is defined in hw1.py as **"start_states"**
6. **F** is the set of final states which are string formations that can be placed **right after** the stem of a verb. It is also defined in hw1.py as **"final_states"**

After modeling the NFA, I implemented the stemmer by using this finite automata. Of course, there are plenty of ambiguities in this stemmer, since Turkish verb structure is really complex. To solve those, I considered carefully all the transitions which may lead ambiguities. I realized that there were a lot of mistakes in transitions and I fixed it one by one. However, fixing is not sufficient, because a lexicon is needed to analyze if the word is correct in Turkish. For example, "arar" can be stemmed both as "ara-r" and "ar-ar". If we had lexicon, we could choose the correct one, which is "ara-r". To summarize, although I did many corrections, I couldn't solve all ambiguities, like the ones based on lexicons.

## Part II

To succeed to create a super stemmer, we had many steps to complete. Firstly, we should strip **"çekim ekleri"** away, which are word additions that do not change the meaning of a word. Secondly, we should erase **"kaynaştırma harfleri"** which are extra letters to sustain the sound harmony of a word. Thirdly, we should strip **"yapım ekleri"** away, which are word additions that not only change the meaning of a word, but also change the type of it. For example, "don" is a stem of verb meaning "freeze", "don-uk" is an adjective meaning "freezed". Here, "uk" formation has changed the type of the word. Finally, we should use a good lexicon in carrying out these steps to make our machine stem correctly. To sum up, a full stemmer needs a lot of efforts to be completed.