**Arabic Tagging Project**

14th meeting – 25.08.2020

**What we did**

We fixed the hierarchy of tags, now every time we see the parent and children of each tag and not all the tags we have. We added the option to add a new tag as the son of the selected tag.

In the suggested tag now each tag has a score which is the #times this tag was tagged / #all tags this term has, ordered from biggest to smallest this way the user can select the “most tagged” easily.

Ilai suggested we use definitions as addition or replacement (?) of tags but Ali explained that the definition by itself will not contribute to finding metaphors, we need a set of defined “classes” and a way to measure how far apart they are which is our tags hierarchy and our travels upon it.

Ilai – lets say we have a poem about fire of love, get me all **הקשרים** of fire all of love and lets understand what fire is based on them, we at the fire ones we find its related to temperature and so on, then on love we find feelings… this way we can put each term we have in one of those categories either love or fire and then we’ll see what are the similarities or differences between love and fire and based on that we’ll understand what we have here aka a metaphor or not. – but again how do you categories each term with a category?

**Problem**

All our tags are based on what the user – Ali thinks is correct, if we wanted to have a more “accurate” or universal tag-term we’d get more users and let them all tag the same poem and take “winning” tag.

**A new way to tag**

So far in order to find a metapore we’ve tagged terms – based on previously tagged terms with same root (Ali’s tags) or get definition, get its root then look at those tags, and we’ll tag the term with the most common (one or more?) - and “calculated” how different sequential terms are from one another, if the difference is big we’ll call it a potential metaphor.

In order to make sure our metaphors are “correct” we’d like to add a new way or ways of finding metaphors, suggestions:

1. Matrix – of co-occurrences.
2. Word embedding.