**Dictionaries Annotation Example**

**Step 1 : Create Models**
- Arabic : 238M tweets → Model of 9M word
- French : 48M tweets → Model of 683k word
- English : 1B tweets → Model of 5M word

**Goal**
Creating text normalisation dictionaries using distributed representations of words, known also as "word embedding", applied on Arabic, French and English Languages.

**Step 2 : Extract Similar words**
The measure of cosine distance is used between the vector of standard-form word and the vectors of every other word in the word embedding model.

The antonym of the standard-form word is used to refine the results. For example, the antonyms exclusion eliminates the possibility of extracting the word *inactive* as a similar word to the word *active*.

**Step 3 : Filter**
The method is to find the longest contiguous matching subsequence that contains no different elements.

**Dictionaries Annotation Example**

<table>
<thead>
<tr>
<th>Misspelled</th>
<th>Standard-word</th>
<th>Correction</th>
<th>Normalisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>aiiiiime (loooooove)</td>
<td>aime (love)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>decevra (will disappoint)</td>
<td>decevoir (disappoint)</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>deballer (unpack)</td>
<td>deprimer (depress)</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>منتاز (misspelled excellent)</td>
<td>ممتاز (excellent)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>اكره (hate him)</td>
<td>اكره (hate)</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>ايمز (dump)</td>
<td>غبي (stupid)</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

**Evaluation of Dictionaries' Content**
- For Arabic langage, an average of :
  - 89.5% in Normalisation success,
  - 83.7% in Correction success.
- For French langage, an average of :
  - 85% in Normalisation success,
  - 73.6% in Correction success.
- For English langage, an average of :
  - 96% in Normalisation success,
  - 86% in Correction success.

**Evaluation of Dictionaries with Sentiment Analysis Tool**
Echo*, an open source software for sentiment analysis based on supervised machine learning algorithm, is used as a test tool. Text Normalisation is applied on both training and testing datasets, of tweets in English language.

Echo f-measure results without Normalisation : 55.64
Echo f-measure results with Normalisation : 56.22

**Future Work**
This work can be a resource for many domains in Natural Language Processing, like Sentiment Analysis and Arabic Dialects Normalisation.

***This work is available as open source on : [https://github.com/amalhtait/NormAFE](https://github.com/amalhtait/NormAFE) & [https://github.com/OpenEdition/NormAFE](https://github.com/OpenEdition/NormAFE) & [http://amalhtait.com/tools.html](http://amalhtait.com/tools.html)***