# A Graph Based Approach for Contextual Text Normalization

Çağıl Uluşahin Sönmez , Arzucan Özgür

Dec 2013

# Introduction

- What is this thing called noise?
- There is a evolving internet language, that has its own slang.
- It is individual.
- The mistakes are also individual.
- The equation is even more complicated when mobile devices get involved.

# Text Normalization

- Text normalization is a preprocessing step to restore **noisy** words in text to their original (canonical) forms.

- The normalization task restores Out of Vocabulary(OOV) words into their In Vocabulary(IV) forms.

- Yet not all OOV word requires normalization: ill-formed words

| | |
|---|---|
| talk 2 u later | talk to you later |
| enormoooos | |
| enrmss | enormous |
| enourmos | |
| ppl | people |
| tanks | tanks, thanks |
| btw | by the way |

**OOV -> IV**

# Choudhury et al. Error Classification

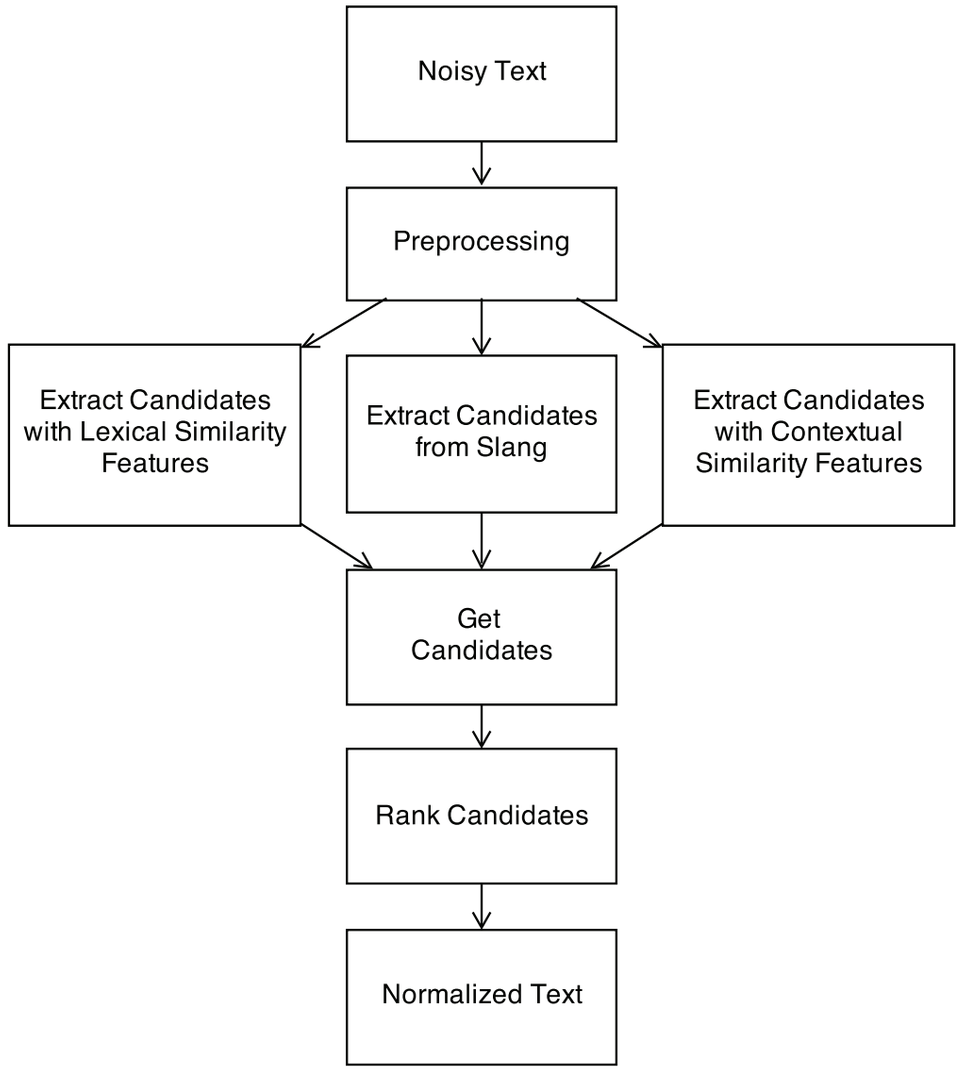
- Unintentional Errors
  - pressing of the wrong key
  - pressing of a key more than the desired number of times
  - deletion of a character
  - inadequate knowledge of spelling
- Intentional Errors
  - character deletion (“tlk” for “talk”, “msg” for “message”, “tomoro” for “tomorrow”, “mob” for “mobile”)
  - phonetic substitution (“nite” for “night”, “bk” for “back”, “u” for “you”, “m8” for “mate”)
  - abbreviations (“btw” for “by the way”, “kgp” for “Kharagpur”)
  - non-standard usage (“wanna” for “want to”, “betta” for “better”, “sumfin” for “something”, “b/c” for “because”)

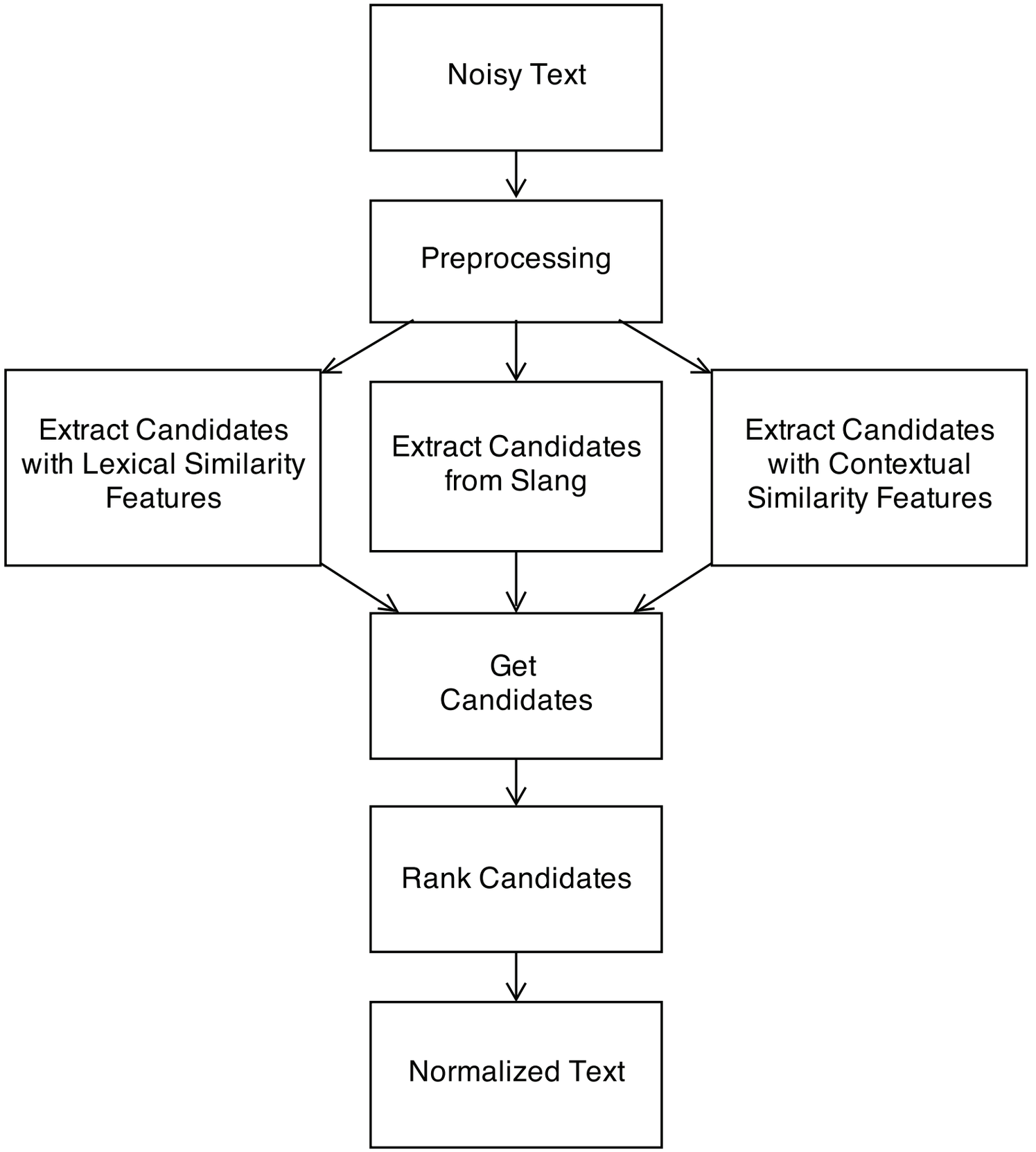
# Related Work

- Noisy channel models
- Lexicon models
- ML systems
- Graph based approaches

# Our Methodology

- A graph based approach
- models both contextual and lexical similarity features
- contextual features which are extracted from a pre-generated directed word association graph
- Lexical similarity features are based on edit distance, longest common subsequence ratio, and double metaphone distance.
- A slang dictionary is used as an external resource to enrich the normalization candidate set.




Noisy Text
Preprocessing
Extract Candidates with Lexical Similarity Features
Extract Candidates from Slang
Extract Candidates with Contextual Similarity Features
Get Candidates
Rank Candidates
Normalized Text

# Preprocessing

| Token | POS tag | Confidence |
|---|---|---|
| with | P | 0.9963 |
| a | D | 0.998 |
| beautiful | A | 0.9971 |
| smile | N | 0.9712 |

| Token | POS tag | Confidence |
|---|---|---|
| w | P | 0.7486 |
| a | D | 0.9920 |
| beatiful | A | 0.9733 |
| smile | N | 0.9806 |

Table 2: Sample POS tagger output obtained by using CMU Ark Tagger (P:Pronoun, D:Determiner, A:Adjective, N:Noun, G:Miscellaneous) [14, 15]

- Tokenization
- Part-of-Speech(POS) tagging

# Word Association Graph

- Contextual information is modeled through a word association graph.
- Created using a large corpus of social media text.
- Directed, weighted graph
- The graph encodes the relative positions of the POS tagged words in the text with respect to each other.
- After preprocessing, each text message in the corpus is traversed in order to extract the nodes and the edges of the graph.

Let's$_L$ start$_V$ this$_D$ morning$_N$ w$_P$ a$_D$ beatiful$_A$ smile$_N$.$_,$

| | |
|---|---|
| Tokens | Let's, start, this, morning, w, a, beatiful, smile, . |
| Nodes | Let's\|L, start\|V, this\|D, morning\|N, w\|P, a\|D, beatiful\|A, smile\|N, .\|, |
| Edges | {Let's\|L, start\|V , distance:1},{Let's\|L, this\|D, distance:2},<br>...<br>{a\|D, beatiful\|A, distance:1}, {a\|D, smile\|N, distance:2},<br>{beatiful\|A, smile\|N, distance:1} |

Table 3: Sample tokenized, POS tagged sentence and the corresponding nodes and edges in the word association graph.

- Each node is a unique set of a token and its POS tag.

- This helps us to identify the candidate IV words for a given OOV word by considering not only lexical and contextual similarity, but also grammatical similarity in terms of POS tags.

- Node properties: *id, oov, freq, tag*.

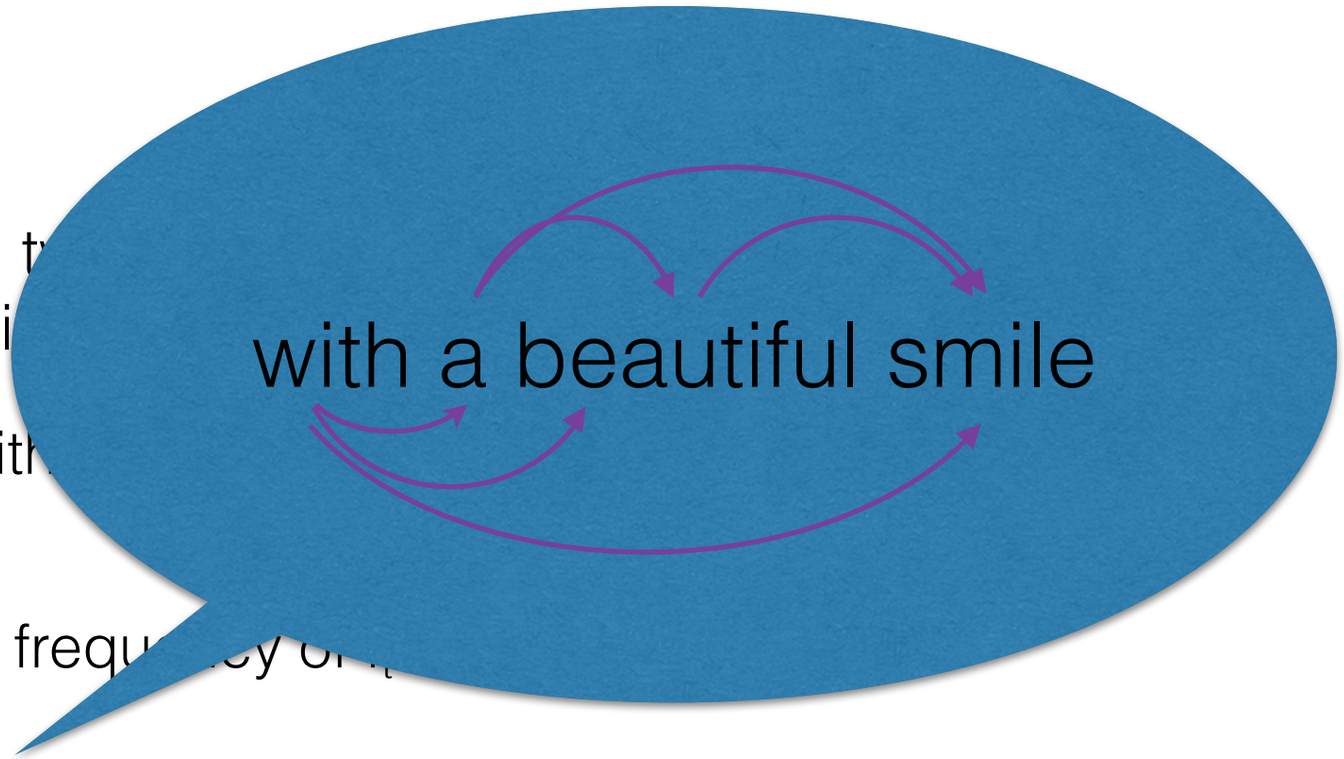
node id : smile|A , freq : 3, oov : False, tag : A

node id : smile|N , freq : 3403, oov : False, tag : N

node id : smile|V , freq : 2796, oov : False, tag : V

Table 4: The nodes in the word association graph representing the token *smile* tagged with different POS tags.

- An edge is created between t
  word pair (i.e. token/POS pai
  - The two words co-occur with
    message in the corpus.
  - Each word has a minimum frequ cy of




- The directionality of the edges is based on the sequence of words in the text messages in the corpus.
- The from property indicates the first word and to is the latter in the phrase.
- The direction and the distance together represent a unique triplet.

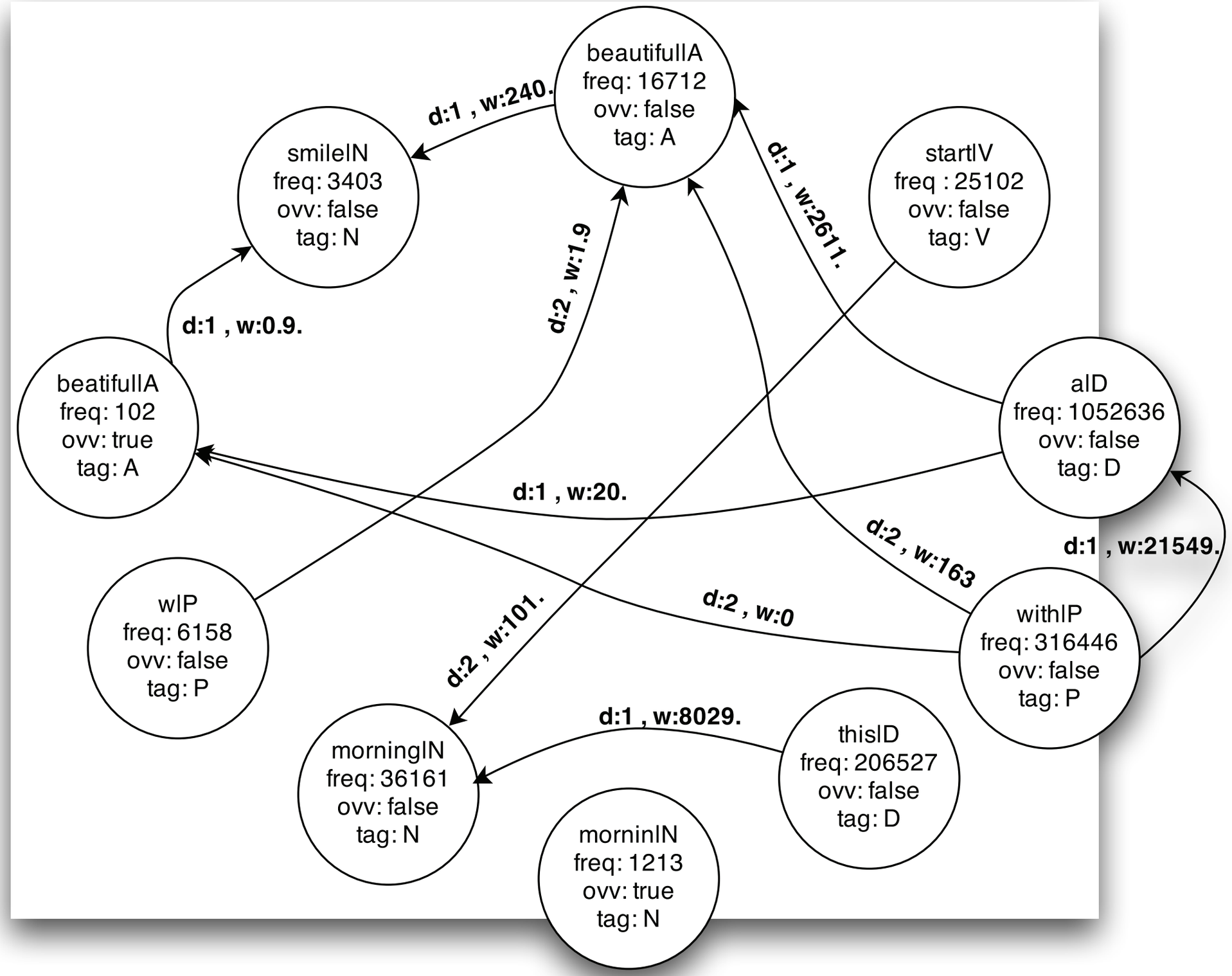
from : with|P, to : smile|N, dis : 3, weight : 72.24415

from : a|D, to : smile|N, dis : 2, weight : 274.37365

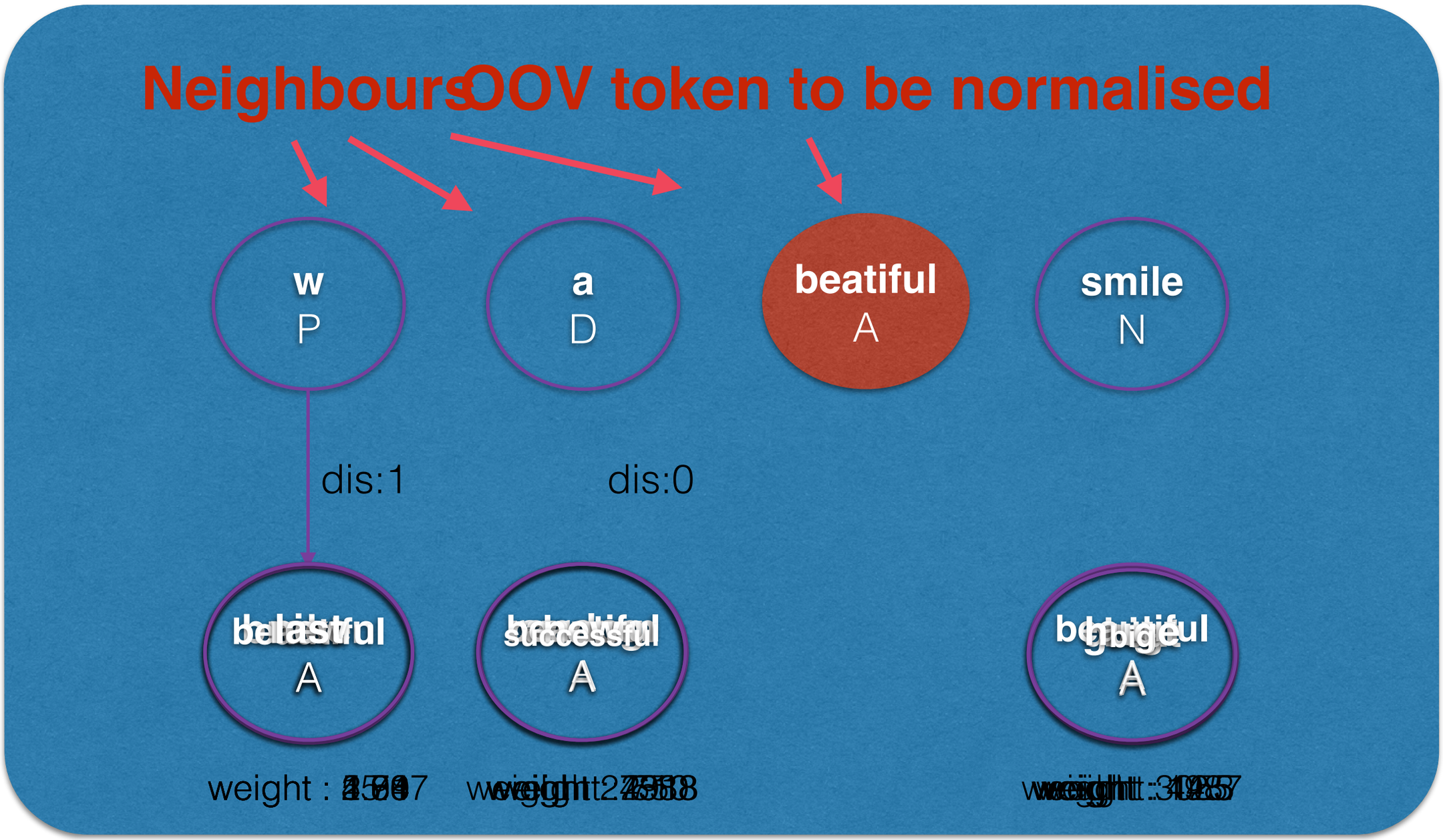
from : beautiful|A, to : smile|N, dis : 1, weight : 240.716

Table 5: Example edges extracted from the sample phrase “with a beautiful smile”

# The Graph

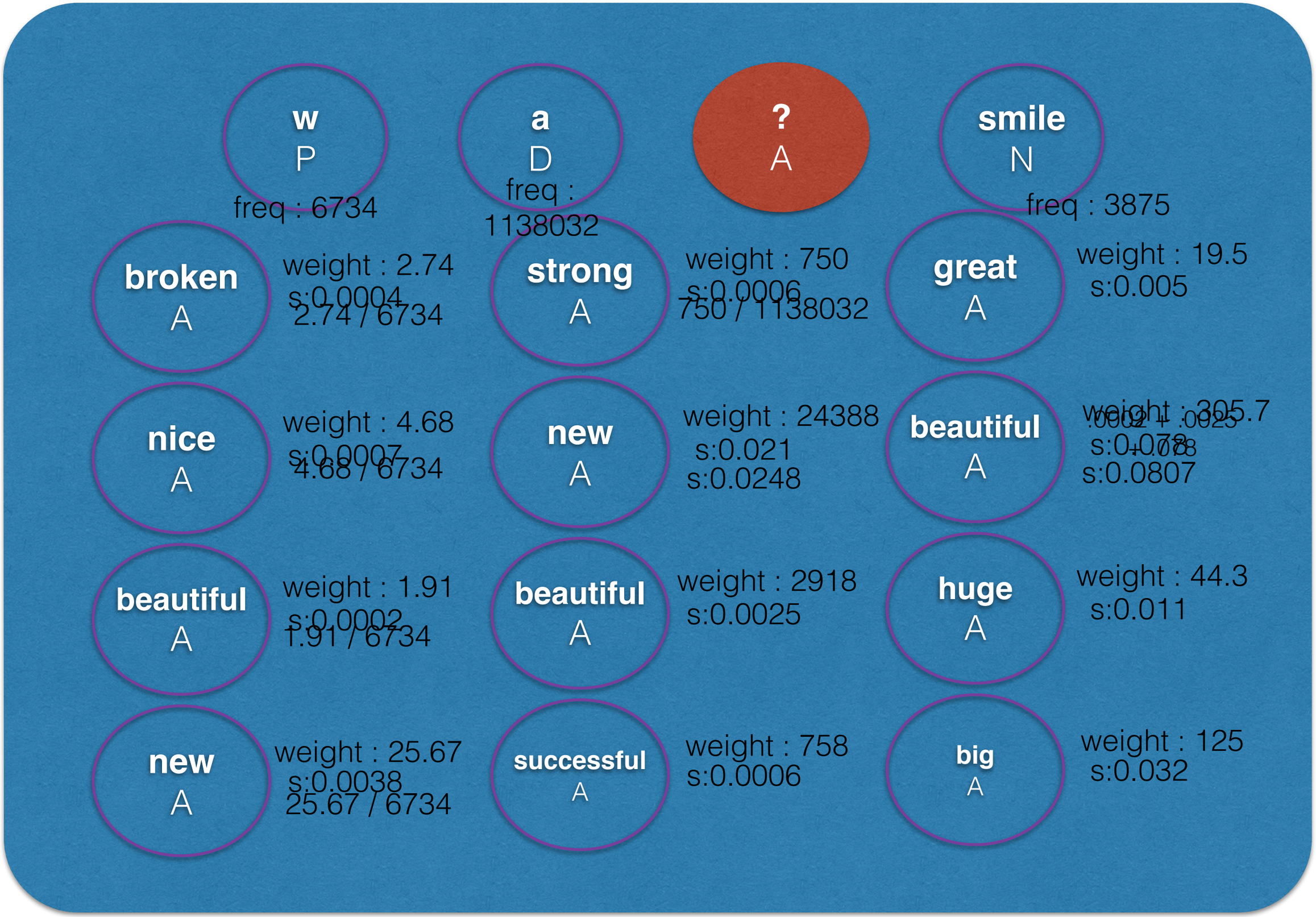

beautifullA
freq: 16712
ovv: false
tag: A
d:1 , w:240.
smileIN
freq: 3403
ovv: false
tag: N
d:1 , w:2611.
startIV
freq : 25102
ovv: false
tag: V
d:1 , w:0.9.
d:2 , w:1.9
beatifullA
freq: 102
ovv: true
tag: A
aID
freq: 1052636
ovv: false
tag: D
d:1 , w:20.
d:2 , w:163
d:1 , w:21549.
wIP
freq: 6158
ovv: false
tag: P
d:2 , w:0
withIP
freq: 316446
ovv: false
tag: P
d:2 , w:101.
d:1 , w:8029.
morningIN
freq: 36161
ovv: false
tag: N
thisID
freq: 206527
ovv: false
tag: D
morninIN
freq: 1213
ovv: true
tag: N

# Graph Based Contextual Similarity



# Contextual Similarity Metrics

- Edge Weight Score
  - The edge weight score, favors and identifies the candidates which are (1) related to many neighbors, and (2) have a high association score with each neighbor.
- Frequency Score
  - The frequency score of the candidate is a real number between 0 and 1. It is proportional to the frequency of the candidate with respect to the frequencies of the other candidates in the corpus.


w
P
freq : 6734
a
D
freq : 1138032
?
A
smile
N
freq : 3875
broken
A
weight : 2.74
s:0.0004
2.74 / 6734
strong
A
weight : 750
s:0.0006
750 / 1138032
great
A
weight : 19.5
s:0.005
nice
A
weight : 4.68
s:0.0007
4.68 / 6734
new
A
weight : 24388
s:0.021
s:0.0248
beautiful
A
weight : 305.7
.0002 + .0025
s:0.078
s:0.0807
beautiful
A
weight : 1.91
s:0.0002
1.91 / 6734
beautiful
A
weight : 2918
s:0.0025
huge
A
weight : 44.3
s:0.011
new
A
weight : 25.67
s:0.0038
25.67 / 6734
successful
A
weight : 758
s:0.0006
big
A
weight : 125
s:0.032

# Candidates with edge weight score and frequency

# Lexical Similarity

- The lexical similarity features are based on edit distance, double metaphone (phonetic edit distance), and longest common subsequence ratio (LCSR)

- We use these features (1) to filter the candidates (2) to find new candidates (3) to score/rank/sort the candidates

# Edit distance & LCSR Scoring

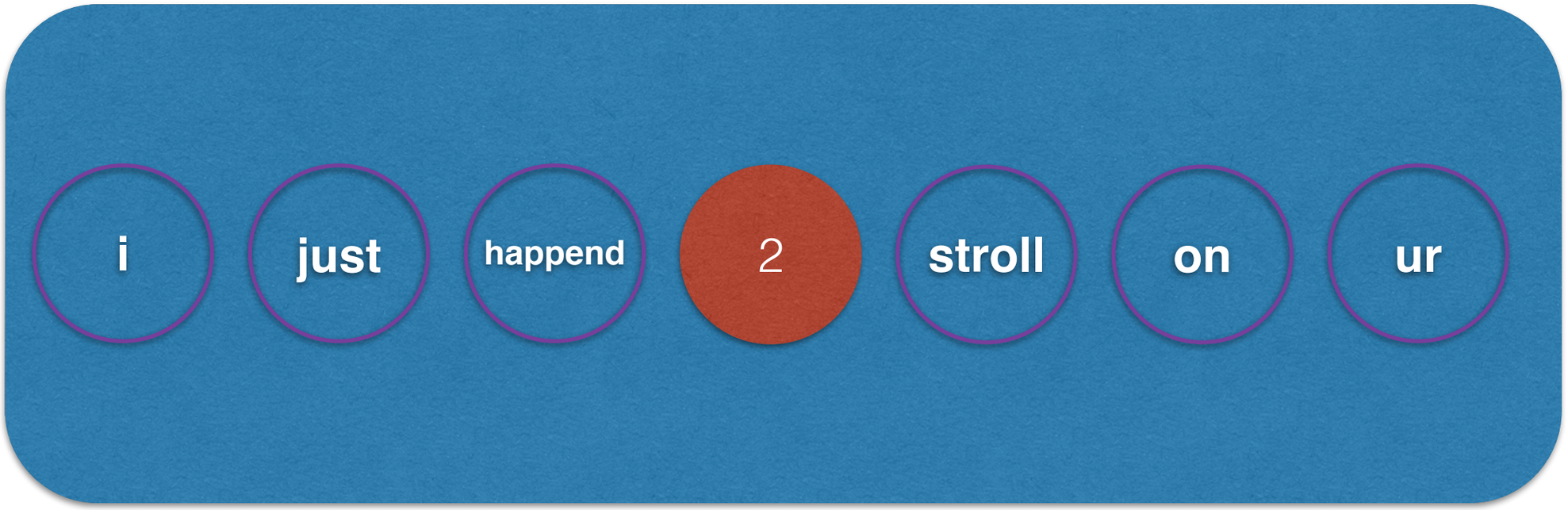
- The edit distance, phonetic edit distance, and LCSR between each candidate and the OOV token are calculated.
- Any candidate with an edit distance greater than edt and phonetic edit distance greater than pedt has been removed from the candidate list
- LCSR is calculated using LCS/max_length*ED_skeleton
- LCSR and Edit distance score are used as lexical features

| OOV | Candidate | LCSR | Distance | Phonetic |
|---|---|---|---|---|
| missin (MSN) | missing (MSNK) | 0.8571 | 0.8572 | 1 |
| missin (MSN) | missed (MST) | 0.6667 | 0.6666 | 1 |
| confrims (KNFR) | confirms (KNFR) | 0.8750 | 0.75 | 0 |
| confrims (KNFR) | confirm (KNFR) | 0.7500 | 0.6240 | 0 |
| soemthing (SMTN) | something (SMTN) | 0.8889 | 0.7778 | 0 |
| soemthing (SMTN) | sorting (SRTN) | 0.6666 | 0.6666 | 1 |

# Window size

- The window size is chosen as 7, with 3 neighbours in each side of the OOV token. (when available)

- Ex: "*I just happend 2 stroll on ur name saw a twit pic I liked so w not u know keep it beautiful : ) ? ? thank u !*"



- Unlike the regular POS taggers designed for well-written newswire-like text, social media POS taggers pro- vide a broader set of tags specific to the peculiarities of social text [14, 15]. Using this extended set of tags we can identify tokens such as discourse mark- ers (e.g. rt for retweets, cont. for a tweet whose content follows up in the coming tweet) or URLs. This enables us to better model the context of the words in social media text.