# Using News Tweets to Generate Predicate Paraphrase Resource

## IST 664 - Natural Language Processing ( NLP )

Ayush Pramod Kumar, Dhaval Sonavaria, Vaibhav Kumar, Rahul Rathod

## Introduction

We surveyed two existing paraphrase resources ( DIRT and Berant ) .It is inferred that the existing resources are not being updated regularly. We, therefore present an approach which guarantees that the resource will be constantly updated. Since, we are querying news tweets on a daily basis to generate binary paraphrase pairs.

## Assumptions

Main assumption: excess news features of a similar occasion are probably going to depict it with various words [3].

This work: recommendations removed from tweets talking about news occasions, distributed around the same time, that concede to their contentions, are predicate reworks.

## Survey of Existing Resources

1. DIRT (10 millions)
Similarities between predicate pairs are estimated by the geometric mean of similarities between arguments.
2. Berant (52 millions)
Matching arguments and applying global optimization (avoiding repetition) to release entailment rules.
3. PPDB (140 million)
Paraphrases extracted from bilingual corpuses scored heuristically.

## Data Source

We query the Twitter Search API via Twitter Search. We use Twitter's news filter that retrieves tweets containing links to news websites, and limit the search to English tweets.



## Approach and Methodology

**Collecting News Tweets**

Using Twitter Search API

get_tweets(lang=en, filter=news) -> Twitter Search -> clean_tweets()

**Proposition Extraction**

Extract propositions from the tweets using PropS [2]
https://github.com/gabrielStanovsky/props

Get binary verbal predicate templates, and apply argument reduction [1]

**Generating Paraphrase Instances**

We think about two predicates as rewords if: (1) They show up around the same time. (2) Every one of their contentions lines up with a one of a kind contention in the other predicate.

Two dimensions of contention coordinating: strict (accurate match/ short alter remove) and free (halfway token coordinating/WordNet equivalent words)

**Generating Types**

$$p_1 = [a]_0 \textbf{ buy } [a]_1, \quad p_2 = [a]_0 \textbf{ acquire } [a]_1$$

$$s(p_1, p_2) = count(p_1, p_2) \cdot \left(1 + \frac{days(p_1, p_2)}{N}\right)$$

- count(p1 , p2 ) assigns high scores for frequent paraphrases
- N - number of days since the resource collection begun
- days(p1,p2) eliminates noise from two arguments participating in different events on the N same day, e.g.:
1) Last year when Chuck Berry turned 90;
2) Chuck Berry dies at 90

**Resource Release**

- We release our resource daily, with two files:
– Instances: predicates, arguments and tweet IDs.
– Types: predicate paraphrase pair types ranked in a descending order according to the heuristic accuracy score.

## Result & Conclusion

| | | | |
|---|---|---|---|
| {a0} approve {a1} | {a0} pass {a1} | 11569 | 544 |
| {a0} meet with {a1} | {a0} meet {a1} | 11017 | 469 |
| {a0} say via {a1} | {a0} say {a1} | 8233 | 715 |
| {a0} kill {a1} | {a0} shoot {a1} | 8142 | 601 |
| {a0} ask {a1} | {a0} tell {a1} | 7181 | 665 |
| {a0} tell {a1} | {a0} urge {a1} | 6892 | 645 |
| {a0} get {a1} | {a0} have {a1} | 6457 | 695 |
| {a0} have {a1} | {a0} take {a1} | 6052 | 718 |
| {a0} tell {a1} | {a0} warn {a1} | 6227 | 632 |
| {a0} get {a1} | {a0} receive {a1} | 6143 | 650 |
| {a0} take {a1} | {a0} win {a1} | 6383 | 501 |
| {a0} announce {a1} | {a0} unveil {a1} | 5460 | 473 |
| {a0} get {a1} | {a0} sentence to {a1} | 5635 | 415 |
| {a0} hit {a1} | {a0} strike {a1} | 5516 | 397 |
| {a0} call {a1} | {a0} slam {a1} | 4713 | 548 |
| {a0} blast {a1} | {a0} slam {a1} | 4479 | 587 |
| {a0} accuse {a1} | {a0} slam {a1} | 4558 | 550 |
| {a0} hit {a1} | {a0} reach {a1} | 4488 | 526 |
| {a0} ask {a1} | {a0} urge {a1} | 4238 | 538 |

1. The above snapshot of our resultant data displays the binary paraphrases.
2. The paraphrases are ranked based upon number of instances, number of different days of their occurrences, number of days since the resource collection began.
3. Our unsupervised method involves reading news tweets discussing the same event to obtain predicate paraphrases.

## Future Scope

Since we generate a large number of paraphrase pairs we sort them into four bins of increasing accuracy the smallest being the most accurate.

Implement supervised learning, check paraphrase pairs before publishing paraphrase source

## References

[1] Gabriel Stanovsky and Ido Dagan. Annotating and predicting non-restrictive noun phrase modifications. In ACL, 2016.
[2] Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. Getting more out of syntax with props. arXiv, 2016.
[3] Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. Automatic paraphrase acquisition from news articles. In HLT, pages 313–318. Morgan Kaufmann Publishers Inc., 2002.