SemEval 2015
Task 1

Paraphrase and Semantic Similarity in Twitter

Task Description

Task A – Paraphrase Identification

Given two sentences, determine whether they express the same or very similar meaning. Following the literature on paraphrase identification, we evaluate system performance by the F-1 score (harmonic mean of precision and recall) against human judgements.

Paraphrase and Semantic Similarity in Twitter

Paraphrase?	Sentence 1	Sentence 2
yes	Ezekiel Ansah wearing 3D glasses wout the lens	Wait Ezekiel ansah is wearing 3d movie glasses with the lenses knocked out
yes	Marriage equality law passed in Rhode Island	Congrats to Rhode Island becoming the 10th state to enact marriage equality
yes	Aaaaaaaaand stephen curry is on fire	What a incredible performance from Stephen Curry
no	Finally saw the Ciara body party video	ciara s Body Party video is on point
no	Now lazy to watch Manchester united vs arsenal	Early lead for Arsenal against Manchester United

Table 1: Representative examples from PIT-2015 Twitter Paraphrase Corpus

Data

	# Unique Sent	# Sent Pair	# Paraphrase	# Non-Paraphrase	# Debatable
Train	13231	13063	3996 (30.6%)	7534 (57.7%)	1533 (11.7%)
Dev	4772	4727	1470 (31.1%)	2672 (56.5%)	585 (12.4%)
Test	1295	972	175 (18.0%)	663 (68.2%)	134 (13.8%)

- Very representative data with many edge cases
- Lots of irregularities
  - This is to be expected from Twitter based data

Baseline Modifications

- Excluding trending name from Tweets
- Tried text normalization which had a negative effect
  - TTYL = “talk to you later”
- Matched given python implementation with an F-Score of .54

Features Overview

- Ark Tweet NLP
- SentiWordNet
- Harvard General Inquirer
- MPQA Subjectivity Lexicon
- Wordnet Synonym
- Chat Speak Translator

Project Achievements

- Uncovered that sentiment is horrible for paraphrase identification
- Normalizing text does not always improve your results
- At some point throwing new features at a problem stops improving results

Summary

- If we could start this project over…
  - Find project that started in Java
  - More research time for our specific problem
  - Make better use of implemented features not brute forcing them

Our Results

feature	train accuracy	train precision	train recall	train F1
best	**0.800**	**0.760**	**0.616**	**0.680**
base	0.790	0.744	0.601	0.665
mod	0.792	0.763	0.578	0.658
ark	0.683	0.592	0.277	0.377
harvard	0.688	0.614	0.266	0.371
sentiwordnet	0.674	0.661	0.121	0.205
subjective	0.673	0.667	0.113	0.193
wordnet	0.788	0.743	0.596	0.661
harvard & wordnet	0.694	0.648	0.256	0.367

Table 5: Our final **training set** results for single layer neural network with softmax

feature	dev accuracy	dev precision	dev recall	dev F1
best	**0.757**	**0.761**	**0.457**	**0.571**
base	0.735	0.746	0.384	0.507
mod	0.751	0.771	0.424	0.547
ark	0.678	0.612	0.254	0.358
harvard	0.670	0.601	0.208	0.309
sentiwordnet	0.644	0.485	0.033	0.062
subjective	0.646	0.509	0.074	0.129
wordnet	0.745	0.748	0.422	0.540
harvard & wordnet	0.652	0.634	0.161	0.248

Table 6: Our final **development set** results for single layer neural network with softmax

SemEval 2015 Task 1&2 Results

TeamRank		TEAM	RUN	task 1 - Paraphase Identification				task 2 - Semantic Similarity				
task 1	task 2			Rank-F1	F1	Precision	Recall	Rank-Pearson	Pearson	maxF1	mPrecision	mRecall
1		ASOBEK	01_svckernel	1	0.674	0.680	0.669	18	0.475	0.616	0.732	0.531
	8	ASOBEK	02_linearsvm	2	0.672	0.682	0.663	14	0.504	0.663	0.723	0.611
2	1	MITRE	01_ikr	3	0.667	0.569	0.806	1	0.619	0.716	0.750	0.686
3		ECNU	02_nnfeats	4	0.662	0.767	0.583					
4		FBK-HLT	01_voted	5	0.659	0.685	0.634	19	0.462	0.607	0.551	0.674
5		TKLBLIIR	02_gs0105	5	0.659	0.645	0.674					
		MITRE	02_bieber	7	0.652	0.559	0.783	2	0.612	0.724	0.753	0.697
6		HLTC-UST	02_run2	7	0.652	0.574	0.754	6	0.545	0.669	0.738	0.611
	3	HLTC-UST	01_run1	9	0.651	0.594	0.720	5	0.563	0.676	0.697	0.657
		ECNU	01_mlfeats	10	0.643	0.754	0.560					
7	4	AJ-SEVAL	01_first	11	0.622	0.523	0.766	7	0.527	0.642	0.571	0.731
8	5	DEPTH	02_modelx23	12	0.619	0.652	0.589	8	0.518	0.636	0.602	0.674
9	9	CDTDS	01_simple	13	0.613	0.547	0.697	15	0.494	0.626	0.675	0.583
		CDTDS	02_simplews	14	0.612	0.542	0.703	16	0.491	0.624	0.589	0.663
		DEPTH	01_modelh22	15	0.610	0.647	0.577	13	0.505	0.638	0.642	0.634
	10	FBK-HLT	02_multilayer	16	0.606	0.676	0.549	17	0.480	0.604	0.504	0.754
10		ROB	01_all	17	0.601	0.519	0.714	10	0.513	0.612	0.721	0.531
11		EBIQUITY	01_run	18	0.599	0.651	0.554					
		TKLBLIIR	01_gsc054	19	0.590	0.461	0.817					
		EBIQUITY	02_run	19	0.590	0.646	0.543					
		BASELINE	**logistic reg**	**21**	**0.589**	**0.679**	**0.520**	**11**	**0.511**	**0.601**	**0.674**	**0.543**
12	11	columbia	02_ormf	22	0.588	0.593	0.583	20	0.425	0.599	0.623	0.577
13	12	HASSY	01_train	23	0.571	0.449	0.783	22	0.405	0.645	0.657	0.634
14		RTM-DCU	01_PLSSVR	24	0.562	0.859	0.417	4	0.564	0.678	0.649	0.709
		columbia	01_ormf	25	0.561	0.831	0.423	20	0.425	0.599	0.623	0.577
		HASSY	02_traindev	25	0.551	0.423	0.789	22	0.405	0.629	0.648	0.611
	2	RTM-DCU	02_SVR	27	0.540	0.883	0.389	3	0.570	0.693	0.695	0.691
		BASELINE	**WTMF**	**28**	**0.536**	**0.450**	**0.663**	**26**	**0.350**	**0.587**	**0.570**	**0.606**
	6	ROB	02_all	29	0.532	0.388	0.846	9	0.515	0.616	0.685	0.560
	7	MATHLING	02_twimash	30	0.515	0.364	0.880	11	0.511	0.650	0.648	0.651
15		MATHLING	01_twiemb	30	0.515	0.454	0.594	27	0.229	0.562	0.638	0.503
16		YAMRAJ	01_google	32	0.496	0.725	0.377	25	0.360	0.542	0.502	0.589
17		STANFORD	01_vs	33	0.480	0.800	0.343					
		AJ-SEVAL	02_second	34	0.477	0.618	0.389					
	13	YAMRAJ	02_lexical	35	0.470	0.677	0.360	24	0.363	0.511	0.508	0.514
18		WHUHJP	02_whuhjp	36	0.425	0.299	0.731					
		WHUHJP	01_whuhjp	37	0.387	0.275	0.651					
		BASELINE	**random**	**38**	**0.266**	**0.192**	**0.434**	**28**	**0.017**	**0.350**	**0.215**	**0.949**