

Detecting metaphors involving motion verbs in online business news sources

CS 182 Project Report
Ambuj Tewari (cs182-cs)

Abstract

The KARMA system, which we saw in class, makes use of metaphorical mappings in trying to understand newspaper stories. In particular, motion verbs play a significant role in conveying facts about entities in the economic or business domain. It is claimed that mappings between the embodied motion domain and the economic one occur frequently in business articles.

This project had two aims: first, to verify the claim that metaphoric use of motion verbs in reporting business stories is a frequent phenomenon, and second, to try using a supervised machine learning approach to classify occurrences of motion verbs in sentences as being metaphoric or not. For articles in the business section of an online newspaper, we saw that 40% of all occurrences of motion verbs were metaphoric and that an off-the-shelf machine learning tool gave about 70% accuracy on the task of classifying unseen instances of motion verbs.

Although we were constrained to work on a small sample size (500 sentences) due to the need to hand label the verb occurrences, these results support, if not verify, the claim that metaphors are used quite frequently in the domain under consideration. They also indicate that it may be possible to use machine learning to automatically classify motion verb instances as metaphoric and non-metaphoric.

1. Introduction

Consider the following sentence from an online business article at San Francisco Chronicle's website¹:

*“Stunned by the Internet implosion, the **stock markets** had been **sliding** for two straight years , the first time that had happened in a quarter century.”*

Here, “stock markets” are entities in the economic domain but the fact that their situation had been steadily deteriorating for a long time is conveyed by the use of the motion verb “sliding”. How is it that we can infer the poor performance of the stock markets from this sentence even when there is no explicit reference to it ? It is by use of two metaphors: one in which the economic entity is viewed as an agent capable of moving or being moved, and the other which establishes a mapping between height and performance (“Higher is Better”).

Now consider this sentence, from the same source:

*“ ‘We 're talking about **moving** in with my mother in San Mateo, an aunt in New York or a cousin in Canada,’ he said. ”*

¹ <http://www.sfgate.com/>

This time, the motion verb “moving” is not used in a metaphoric sense but refers to an actual movement of people. In this report we will explore whether it is possible to have a program learn to make this distinction.

Lexicon building efforts, like FrameNet, completely ignore metaphors. Given the widespread use of metaphors in newspapers, etc., we ought to build systems that can understand and analyze metaphoric sentences. But before we do that, we need a corpus of sentences containing words used in a metaphoric sense. If it possible to have an automated way of extracting metaphoric sentences from articles on the web, we can easily build such a corpus using the numerous articles available online in newspaper archives.

This report describes a small effort in that direction and is organized as follows. The next section will describe the design of the system built as part of this project. We describe the learning technique and results obtained in Section 3. Possible extensions to this work are discussed in Section 4.

2. System architecture

The task of learning to recognize metaphoric instances of motion verbs is accomplished in two stages. First, sentences containing motion verbs are extracted from an online source automatically. A human needs to hand label the motion verb occurrences as metaphoric and non-metaphoric. We provide a GUI application to ease this task a little bit. A classifier is then learned from these labeled examples using a Support Vector Machine (SVM) package. In the next stage, we extract more sentences from the web and use the learned classifier to automatically classify them. See figure 1 for a schematic of the system architecture.

We used two well tools in this project: Charniak’s parser² for parsing the sentences and LIBSVM³ – a library for SVM’s. The rest of the code was written in Python, an interpreted, platform independent, object-oriented scripting language ideal for rapid prototyping projects such as this one.

Various components of the system are briefly described in the subsections below.

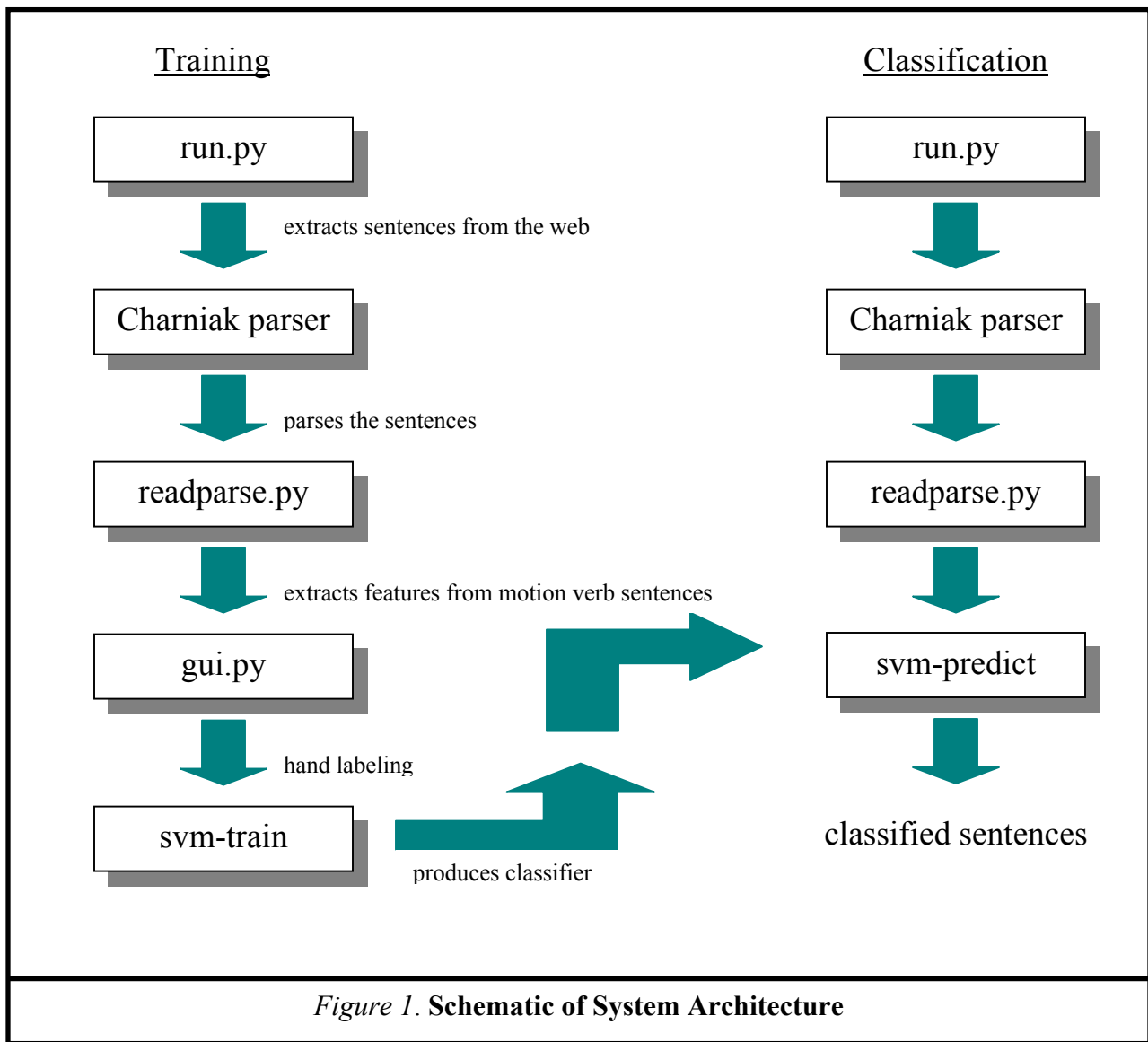
2.1 run.py

This is a Python script that downloads HTML pages from the news source specified on the command line. Python provides extensive library support for connecting to URL’s and parsing HTML documents. This makes it fairly straightforward to extract text from the online pages. The script then saves the sentences in a form that can be given as input to Charniak’s parser.

At present only two news sources are supported: The Hindu and San Francisco Chronicle, but it is easy to add more.

² <ftp://ftp.cs.brown.edu/pub/nlparser/>

³ <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>



2.2 Charniak's parser

This is a statistical parser for parsing English text and produces tagged parse trees for the input sentences. The tags are the standard tags found in University of Pennsylvania tree-bank tag set⁴.

2.3 readparse.py

This script reads the output of Charniak's parser and uses it to find motion words that are used as verbs in a sentence. For the list of motion words, (generated using the lexical

⁴ <http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html>

database WordNet⁵) see Appendix A. For every motion verb in a given sentence, we look at the two closest nouns appearing its left and to its right. For each of them we find whether it is a proper noun and whether it is a business word. Again, see Appendix B for the list of business words. This gives us a feature vector (having eight Boolean features) per motion verb.

The script stores each motion verb along with the containing sentence and feature vector in a text file.

2.4 gui.py

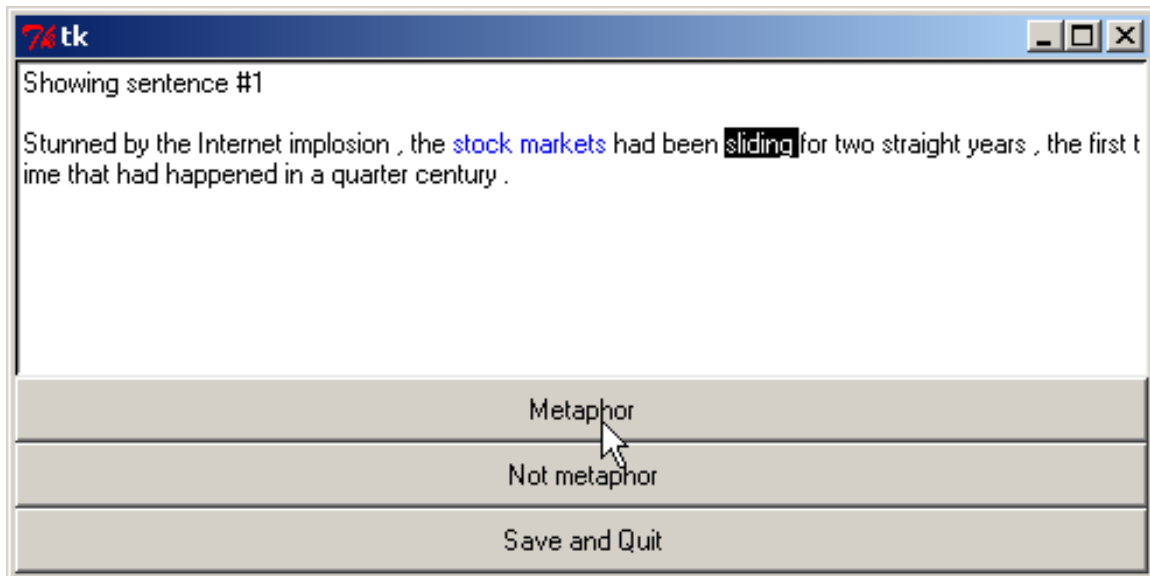


Figure 2. Screenshot of gui.py

Shown in figure 2 is a screenshot of the interface of the script gui.py. It is used to attach labels to the feature vectors computed by readparse.py.

2.5 svm-train

This is a utility that comes with the LIBSVM package mentioned above. It uses SVM's to generate a classifier from a given set of labeled examples. The user can choose from a variety of options for the kernel to be used. The default kernel type is radial basis function and is what we used.

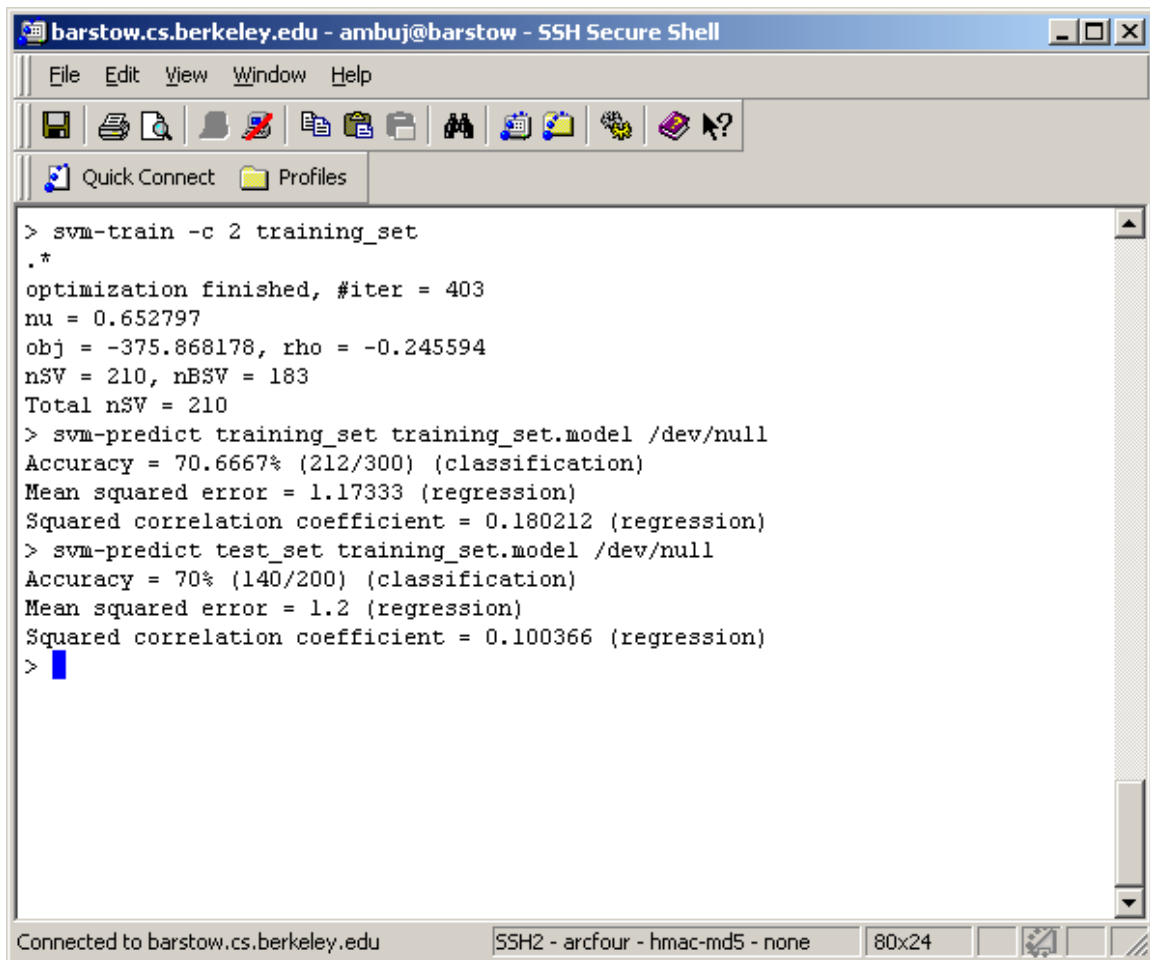
⁵ <http://www.cogsci.princeton.edu/~wn/>

2.6 svm-predict

This is a utility that uses a classifier previously generated by svm-train to classify new examples. It also measures the accuracy of prediction in case the true labels of the examples are known.

3. Results

We took 500 occurrences of motion verbs in parsed sentences and labeled them all manually using gui.py. They were split into a training set of size 300 and a test set of size 200. Figure 3 shows the result of the svm-predict on the training and test sets.



```
barstow.cs.berkeley.edu - ambuj@barstow - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
> svm-train -c 2 training_set
.*
optimization finished, #iter = 403
nu = 0.652797
obj = -375.868178, rho = -0.245594
nSV = 210, nBSV = 183
Total nSV = 210
> svm-predict training_set training_set.model /dev/null
Accuracy = 70.6667% (212/300) (classification)
Mean squared error = 1.17333 (regression)
Squared correlation coefficient = 0.180212 (regression)
> svm-predict test_set training_set.model /dev/null
Accuracy = 70% (140/200) (classification)
Mean squared error = 1.2 (regression)
Squared correlation coefficient = 0.100366 (regression)
>
Connected to barstow.cs.berkeley.edu  SSH2 - arcfour - hmac-md5 - none  80x24
```

Figure 3. Screenshot of svm-train and svm-predict

The SVM classifier achieved accuracies of 70.6667% and 70% on the training and test sets respectively.

After hand labeling 500 examples we found that 213 of them were metaphoric which is about 42% of the total.

4 Discussion and Extensions

The way features are computed right now is rather ad hoc and arbitrary. In fact, it is rather remarkable that such a naive approach achieved 70% accuracy. For example, consider the sentence:

“Dreyer 's Grand Ice Cream, of Oakland, seized second place, climbing 85 percent for the year to hit \$70.96 .”

The closest nouns to the left are “Oakland” and “place” but have nothing to do with the agent of the verb “climbing”. In fact, figuring out the agent of a motion verb is a problem that has not been solved completely.

Also, the classifier can get confused by proper names. In business articles, a proper name often refers to companies, countries, etc. and hence it is reasonable to assume that if it appears close to a motion verb, the agent of the verb is an abstract economic entity. Such an assumption is hardly justifiable in general.

One desirable extension of the project is to design a kernel that captures the relevant features in this problem and use it directly instead of indirectly proceeding via feature vectors and using a standard kernel. To describe the theory of SVM's is beyond the scope of this report but it suffices here to mention that their performance in a given application is heavily dependent on how good the choice of the kernel function is. There has been some recent work on designing kernels of text mining but it was not consulted for this project.

5 Acknowledgements

Many thanks to Professor Feldman for providing the opportunity to do this project. We also indebted to Srinu Narayanan for many hours of helpful discussions.

Appendix A: Motion words

abseil	collapses	flopping	jogging	marched
abseils	collapsing	flopped	jogged	move
abseiling	collapsed	flounce	joggle	moves
abseiled	crash	flounces	joggles	moving
accelerate	crashes	flouncing	joggling	moved
accelerates	crashing	flounced	joggled	nosedive
accelerating	crashed	flow	jolt	nosedives
accelerated	crawl	flows	jolts	overshoot
advance	crawls	flowing	jolting	overshoots
advances	crawling	flowed	jolted	overshooting
advancing	crawled	flutter	journey	overshot
advanced	creep	flutters	journeys	overturn
amble	creeps	fluttering	journeying	overturns
ambles	creeping	fluttered	journeyed	overturning
ambling	crept	fly	jump	overturned
ambled	cruise	flies	jumps	pace
approach	cruises	flying	jumping	paces
approaches	cruising	flew	jumped	pacing
approaching	cruised	gait	kick	paced
approached	dart	gaits	kicks	parachute
ascend	darts	gaiting	kicking	parachutes
ascends	darting	gaited	kicked	parachuting
ascending	darted	gallop	kneel	parachuted
ascended	dash	gallops	kneels	parasail
balloon	dashes	galloping	kneeling	parasails
balloons	dashing	galloped	kneeled	parasailing
ballooning	dashed	glide	knock	parasailed
ballooned	descend	glides	knocks	pitch
block	descends	gliding	knocking	itches
blocks	descending	glided	knocked	itching
blocking	descended	haste	lash	itched
blocked	dive	hasten	lashes	plod
bolt	dives	hastens	lashing	plods
bolts	diving	hastening	lashed	plodding
bolting	dived	hastened	leapfrog	plodded
bolted	dogtrot	heave	leapfrogs	prance
canter	dogtrots	heaves	leapfrogging	prances
canters	dogtrotting	heaving	leapfrogged	prancing
cantering	dogtrotted	heaved	limp	pranced
cantered	drift	hitch	limps	progress
carom	drifts	hitches	limping	progresses
caroms	drifting	hitching	limped	progressing
caroming	drifted	hitched	lope	progressed
caromed	drive	hobble	lopes	promenade
chasse	drives	hobbles	loping	promenades
chasses	driving	hobbling	loped	promenading
chasseing	drove	hobbled	lunge	promenaded
chassed	fall	jaunt	lunges	prowl
circle	falls	jaunts	lunging	prowls
circles	falling	jaunting	lunged	prowling
circling	fell	jaunted	lurch	prowled
circled	flicker	jerk	lurches	push
clamber	flickers	jerks	lurching	pushes
clambers	flickering	jerking	lurched	pushing
clambering	flickered	jerked	maneuver	pushed
clambered	flit	jiggle	maneuvers	ramble
climb	flits	jiggles	maneuvering	rambles
climbs	flitting	jiggling	maneuvered	rambling
climbing	flitted	jiggled	march	rambled
climbed	flop	jog	marches	rap
collapse	flops	jogs	marching	raps

raping	shambles	stoop	treads	wag
raped	shambling	stoops	treading	wags
ride	shambled	stooping	treaded	waging
rides	skid	stooped	trek	waged
riding	skids	stride	treks	waggle
rode	skidding	strides	trekking	waggles
roll	skidded	striding	trekked	wagging
rolls	skip	strode	trip	waggled
rolling	skips	stroll	trips	walk
rolled	skipping	strolls	tripping	walks
run	skipped	strolling	tripped	walking
runs	slide	strolled	trot	walked
running	slides	strut	trots	wallop
ran	sliding	struts	trotting	wallops
rush	slid	strutting	trotted	walloping
rushes	slip	strutted	trudge	waloped
rushing	slips	strumble	trudges	wander
rushed	slipping	stumbles	trudging	wanders
sail	slipped	stumbling	trudged	wandering
sails	speed	stumbled	turn	wandered
sailing	speeds	swing	turns	waver
sailed	speeding	swings	turning	wavers
sashay	speeded	swinging	turned	wavering
sashays	spin	swung	twiddle	wavered
sashaying	spins	tailspin	twiddles	whirl
sashayed	spinning	tailspins	twiddling	whirls
saunter	spun	thrust	twiddled	whirling
sauntering	spiral	thrusts	twirl	whirled
sauntered	spirals	thrusting	twirls	wiggle
scamper	spiraling	thrust	twirling	wiggles
scampers	spiraled	trail	twirled	wiggling
scampering	sprint	trails	twist	wiggled
scampered	sprints	trailing	twists	wind
scramble	sprinting	trailed	twisting	winds
scrambles	sprinted	tramp	twisted	winding
scrambling	squirm	tramps	volley	winded
scrambled	squirms	tramping	volleys	wriggle
scud	squirming	tramped	volleying	wiggles
scuds	squirmed	travel	volleyed	wriggling
scudding	stagger	travels	waddle	wriggled
scudded	staggers	traveling	waddles	zoom
scurry	staggering	traveled	waddling	zooms
scurries	staggered	traverse	waddled	zooming
scurrying	step	traverses	wade	zoomed
scurried	steps	traversing	wades	
shamble	stepping	traversed	wading	
	stepped	tread	waded	

Appendix B: Business words

accountants
accounting
ads
advertising
allocations
amount
audit
average
averages
bank
bankruptcy
bargain
benefits
bill
billing
bonds
bonuses
budget
business
businesses
capital
cash
cent
cents
commodities
commodity
companies
company
compensation
competition
corporate
corporations
cost
costs
counties
countries
country
credit
debt
debts
deficit
demand
discounts
dividend
dividends
dollar
dollars
dot-com
earnings
economy
employment
estimates
expenses
fare
fares
fee
fees
firm
firms
freight
fund

funds
gain
gains
government
income
index
industry
inflation
insurance
interest
investment
layoff
layoffs
loss
losses
market
markets
money
mortgage
mortgages
pay
payments
percent
percentage
premium
premiums
price
prices
production
profit
profits
purchases
rate
rates
rent
rents
sales
share
shares
spending
spendings
stock
stocks
tax
taxation
taxes
transactions
wages
wealth