NLP and Sentiment Driven Automated Trading

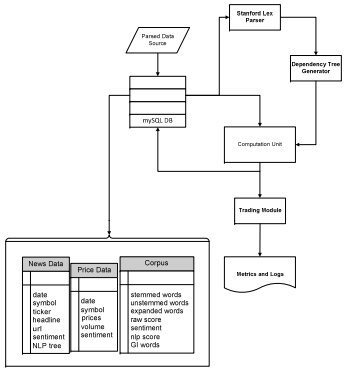
Atish Davda, Parshant Mittal
Faculty Advisor: Michael Kearns

Abstract

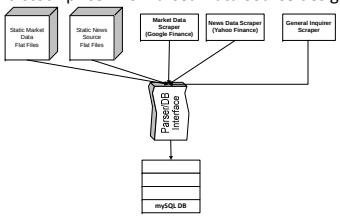
- Perform **syntax based sentence-level analysis on headlines** of stock news, to determine trading feasibility.
  Infer word-sentiments from stock prices to trade stocks.

Final System Design

- Implemented in three phases, incrementally adding functionality; the final system diagram is shown below.



- The *Parsed Data Source*, written in Java for consistency with the remainder of the system, actually relies on Perl scripts which run periodically during the day and scrape **Yahoo! Finance** and **Google Finance** websites and gather news and stock price info. *Parsed Data Source* design:

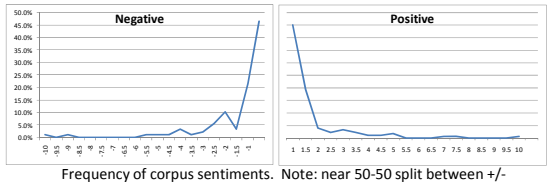


Static Flat Files are remnants of Phase 1 of the project, when we used three months of static news and price data.

- For greatest flexibility, allowing us to **grab headlines** for any given stock for any given date, we parsed news websites.
- Similarly, **retro-parsed price info (open, close, volume)** for any given date, storing both into the MySQL database.
- The natural next step in making the system live is **integrating RSS feeds** into the data source.

Sentiment Generation

- Data was gathered from **July through November 2007**. As you know these were turbulent times for all equities markets, as evident from the *Market Return* chart.
- Daily news and prices from July and August were used as the **"learning bed" to assign sentiments (between -10 to 10) to corpus**.



Frequency of corpus sentiments. Note: near 50-50 split between +/-

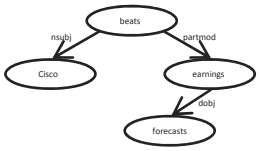
- Corpus
  - **Hand-selected 200 high-frequency words** were "stemmed" and **expanded, using WordNET, to over 2100** altogether.
    Stemming: using the Stanford Lex Parser's stemming algorithm, reduce (*falling*, *falls*, *fall*, *fell*) to (*fall*).

NLP – Dependency Trees

- **Extracted relationships between words**, implied by syntax of headline using the Stanford Lex Parser. Once we classified each word by dependency tags (below) we built dependency trees.

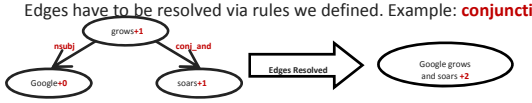
aux – auxiliary
  auxpass - passive auxiliary
  cop - copula
conj - conjunct
cc - coordination
arg - argument
  subj - subject
  nsubj - nominal subject
    nsubjpass - passive nominal
  csubj - clausal subject
comp - complement
  obj - object
    dobj - direct object

Parser's dependency tags

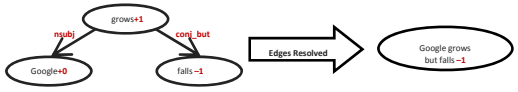


Tree for: "Cisco beats earnings forecasts."

- **Dependency tree structure**: vertices are words, edges are relationships.
- Analysis begins at leaf nodes and then percolating up until the root. Edges have to be resolved via rules we defined. Example: **conjunctions**



"Google grows and soars." Note: the number after the word is it's sentiment.



"Google grows but falls." Note: the number after the word is it's sentiment.

- **Difficult to apply on headlines – NOT sentences.**
  Rules for many dependencies were manually developed by sampling headlines for ideas expressed within.

Trading Strategies & Results

- Various hypotheses were tested, including impact of **no. of headlines, volume of trade, gain/loss** on previous day.
- Predicted sentiment for each stock for each day were computed. **BUY/SELL/HOLD correctness was measured** by taking close-to-close return, i.e. taking a position.

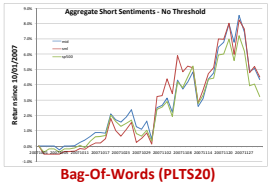
Strategy Name	Description of Decision Making (Opp. To *Sell (Hold* if not *Buy/Sell*))
PLTS10 - Pure Momentum	Buy if stock price gained on previous day.
PLTS20 - Sent. Signal	Buy if sentiment of headline > 0.
PLTS21 - Sent. Signal Threshold	Buy if sentiment of headline > 1 stdev above historical mean sentiment.
PLTS22 - Sent. Expanded	Buy if expanded list of sentiment of headline > 0.
PLTS23 - Sent. Expanded Threshold	If expanded sentiment list > 1 stdev above historical mean sentiment.
PLTS24 - Sent. Respective Index	Buy if this sentiment > 0.
PLTS30 - Sent. Volume	If Δvolume from yesterday is 20% greater, follow rule PLTS20.
PLTS31 - Sent. NewsCount	ratio: sentiment/number of news events. If ratio < small threshold, buy.
PLTS40 - NLP: Sent.	Build dependency tree. If sentiment > 0, buy.
PLTS41 - NLP: G.I.	Build dependency tree with General Inquirer. If GI_sentiment > 0, buy.
PLTS42 - NLP: Sent. & G.I.	Build dependency tree with sentiment and G.I. If this sentiment > 0, buy.
PLTS50 - Market movement.	Overall market.
PLTS60 - Running Avg/StDev	Determine Threshold by keeping running averages/standard deviations.

Trading Strategies Measured

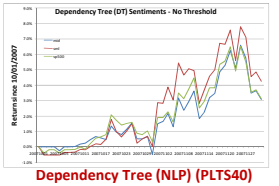
- **Baseline: market returns by index** during Oct, Nov 2007, and **Strategy returns** aggregated over indices:

Market Return

Aggregated Strategy Return



Bag-Of-Words (PLTS20)



Dependency Tree (NLP) (PLTS40)

- Summary: two-month and annualized returns (averaged over the three indexes) for the different strategies.

Return	PLTS10	PLTS20	PLTS21	PLTS22	PLTS23	PLTS24	PLTS30	PLTS31	PLTS40	PLTS41	PLTS42	PLTS50
2-Mo.	-3.5%	4.0%	3.2%	3.1%	3.2%	2.9%	1.0%	3.9%	3.5%	0.6%	3.7%	-8.7%
APY	-19.4%	26.7%	20.5%	20.3%	21.0%	18.6%	6.4%	25.5%	22.9%	3.5%	24.6%	-42.1%

Conclusions

- Even ignoring context, **Bag-of-Words sentiment** analysis yields promising results, **+4.0% (12.7% over market)**.
- Sentiment-free momentum strategy yields -3.5% (i.e. momentum effect not too significant).
- **Dependency analysis yields +3.5%, (12.2% over market)**, but no improvement over Bag-of-Words.
- **NLP NOT INCREMENTALLY EFFECTIVE OVER SENTIMENTS** because headlines do not follow standard grammatical rules.