

A Software Framework for a News Event Driven Simulation of Algorithmic Trading Strategies

Architecture and Simulation Tests

Andreas Storckenmaier, Marius Müller, Christof Weinhardt

*Institute of Information Systems and Management
Information and Market Engineering
Karlsruhe Institute of Technology*

1 Introduction

The fast development of information systems during the last decades has had a significant impact on the financial services industry and the securities trading industry in particular (Jain 2005, Stoll 2006). Automated trading systems have become standard on major exchanges world-wide and algorithmic trading has overtaken human trading in some markets in terms of generated trading volume. However, little academic evidence exists on the effects and profitability of algorithmic trading.

A single accepted definition of algorithmic trading (AT) does not exist but a commonly used definition in the academic world is that AT is “the use of computer algorithms to manage the trading process” (Hendershott et al. 2008). AT can basically be separated into agency algorithmic trading and proprietary algorithmic trading (Hasbrouck and Saar 2009). This separation can also be found in the New York Stock Exchange (NYSE) program trading descriptions. Agency AT essentially comprises the automation of broker tasks which includes the automation of order generation and submission through monitoring of market conditions and splitting of large orders over time and markets to minimize the impact of orders. Proprietary AT comprises of statistical arbitrage, automated market making, and high frequency trading using high speed recognition of specific market patterns and situations.

A relatively new form of proprietary AT is news based proprietary AT which automatically analyzes stock specific and general economic news to trade profitably. Major providers of financial news like Thomson Reuters or Dow Jones have developed machine readable news products that already offer electronically pre-interpreted news. One example is the Reuters NewsScope Sentiment Engine

(RNSE) system¹. Also regulators like the U.S. Securities and Exchange Commission (SEC) increasingly demand from companies to provide regulatory information electronically. SEC regulation requires public companies to submit their annual financial statements in XML based XBRL (eXtensible Business Reporting Language) which allows for an easy dissemination and analysis of financial information².

We present in our paper a Java based software framework that supports the agent based simulation of news based proprietary algorithmic trading strategies and provide first simulation test results. We do not model an artificial stock market but base our analysis on historical trade and quote data and model our agents as price takers in highly liquid markets. In liquid markets orders up to a considerable size trade usually at the best bid and best ask which justifies the price taking assumption. With our framework we are trying to address questions between financial econometrics and pure artificial stock market simulations as we enable researchers to analyze dynamic strategies on real world data. Additionally we are able to model explicit exchange system effects like system latency. The main contribution of the paper is the presentation of the software framework.

Although the major goal of the paper is to present the framework, we also include tests based on existing non-artificial data to show that our framework is implemented and running stable.

The paper is structured as follows. Section 2 discusses relevant finance and financial simulation literature. Section 3 introduces our software framework. Section 4 provides in-depth information about the testing data. Section 5 provides first results of simulation test runs and section 6 finally concludes.

2 Literature Review

The relevant literature in finance consists of two different research streams, the influence of news and information events on financial markets and the specific analysis of algorithmic trading. An overview of relevant finance literature will motivate the development of our software framework.

Analyses of news and financial markets have only appeared recently in the finance literature. Tetlock (2007) analyzes the effect of the Wall Street Journal (WSJ) column 'Abreast of the Market' on the American stock market and the effect of the market on the column. He finds that high pessimism in the WSJ column is followed by lower market prices and thereafter by a reversal to fundamentals. This finding is consistent with noise trader theories and could potentially be exploited through trading strategies. He extracts the pessimism factor using con-

¹ For more information on the Reuters NewsScope Sentiment Engine system see <http://www.thomsonreuters.com/content/financial/pdf/enterprise/NewsScopeSentimentEngine.pdf>

² See <http://www.sec.gov/spotlight/xbrl.shtml>

tent analysis techniques. Tetlock et al. (2008) analyze whether linguistic content comprises information relevant for financial markets. They find that relevant information that would be hard to quantify is contained within linguistic media content. An analysis of the reaction of investors to stale information about S&P 500 firms shows that markets react to stale news through individual overreacting investors but then show subsequent return reversals (Tetlock 2008). Anweiler and Frank (2004) investigate the link between the information content of Internet stock message boards and financial markets. They use Naïve Bayesian analysis and support vector machines to classify message board stories. Their finding is that stock message boards support predicting market volatility. News also effects individual buyers' perception of and their attention towards certain stocks (Barber and Odean, 2008). Individual buyers are more prone to buy stocks which have drawn their attention through media outlets. Additionally, there is empirical evidence that the breadth of information dissemination has an influence on stock returns (Fang and Peress, 2008). All results presented so far are based on daily analyses. Green (2004) studies the intraday effects of economic news releases on U.S. government bond trading. He finds an increase in adverse selection costs after economic news. This indicates that more information is impounded into markets as a result of economic announcements. Apparently there is a significant link between news, media outlets, and the reaction of financial markets. In contrast to most studies our simulation is focused on intraday reactions to news.

Despite its importance for practitioners and its influence on most financial markets there is little academic evidence on the influence of algorithmic trading (AT) on market quality. There is no academic work on the interrelationship between high-frequency news and algorithmic trading. There are two current papers that have direct access to market wide detailed AT data and conduct a thorough and methodologically consistent analysis of AT's influence on financial markets. Chaboud et al. (2009) study AT in the foreign exchange market and its impact on volatility and liquidity. They find that AT does not increase volatility and that, following a macroeconomic announcement, algorithmic traders pull out of the market for a short time and then supply relatively more liquidity than non-AT. Consistently, Hendershott and Riordan (2009) also find no conclusive link between AT and market volatility. However, they find that AT contributes more to the efficient price discovery than non-AT and that AT provides liquidity when it is expensive and consumes when it is cheap. Riordan and Storkenmaier (2009) study an exchange system upgrade and attribute increased liquidity and higher quote based price discovery to speed sensitive algorithmic liquidity supply. Hendershott et al. (2008) use exchange system message traffic as a proxy for AT and find evidence that AT improves liquidity.

To our knowledge there is currently no theoretical market microstructure model which explicitly includes algorithmic trading. The Copeland and Galai (1983) model was developed well before the rise of algorithmic trading. However, it is one model that includes latency, an important issue for AT. The authors view

standing limit orders as options that provide free-trading opportunities. They model the impact of latency on the option value of limit orders and find that a reduction of latency reduces the option value and exposure risk of limit orders. Biais and Weill (2009) develop a model of limit order trading that includes market monitoring and its relation to liquidity. Since algorithms change the determinants of market monitoring this model helps to shed light on algorithmic trading from a theoretical point of view.

Financial market simulation is not an idea that has only emerged during the last years but has a longer history. As early as 1985, Hakansson et al. (1985) develop a simulation to analyze the possibility of automated market making. They base their ideas on the NYSE market model which includes a market maker – at the NYSE called specialist - to smooth supply and demand. Hakansson et al. (1985) simulate a stock market based on artificial demand processes and find that automation of market making with simple rules is in principle possible. Many simulations on financial markets are round based. For example, Brock and Hommes (1997) introduce a round based simulation where traders submit their decision at discrete points in time. They model agents in a completely artificial stock market that adjust their beliefs over time. A time continuous model is suggested by Shatner et al. (2000). Boer et al. (2005) give an overview on the development and pitfalls of artificial stock markets. They compare existing research to real existing stock markets and propose a continuous-time model as opposed to discrete-time models. Boer et al. (2007) then develop an agent based simulation framework that enables continuous-time and asynchronous modeling of financial markets. Their framework is based on artificially generated price processes.

In contrast to existing simulations, we develop a framework that enables researchers and practitioners to simulate time-continuous markets with agents (AT) trading on historical instead of artificially generated price processes. As a special characteristic of our framework, we offer agents a stream of external news or events that have a potential influence on price discovery processes. Our framework allows for an explicit modeling of exchange system infrastructure effects in principal also based on stochastic processes and statistical distributions.

3 Software Design and Architecture in a Nutshell

The software framework is based on a flexible and extensible structure but tailored to some specific requirements that we have for our simulation. First, we need our framework to provide price processes based on historical data. Second, we need an architecture that specifically excludes influences of communication latencies on simulation results, and third we need our framework to be able to handle an external event stream. We extensively use standard design patterns throughout the software development process to ensure a structured, comprehensible, and reusable architecture. Examples of parts where we use design patterns are Mana-

gers and Entities which are based on the manger design pattern and the creation of and access to the Time Emitter which is based on the singleton pattern. The creation of messages is based on the factory design pattern. Our framework is implemented in Java 6 and does not use operating system specific components. We have successfully run test simulations on Microsoft Windows as well as on Linux systems.

The framework consists of two basic layers, the virtual environment and the stock market simulation that is hosted within this virtual environment. Figure 1 gives an overview of the system architecture.

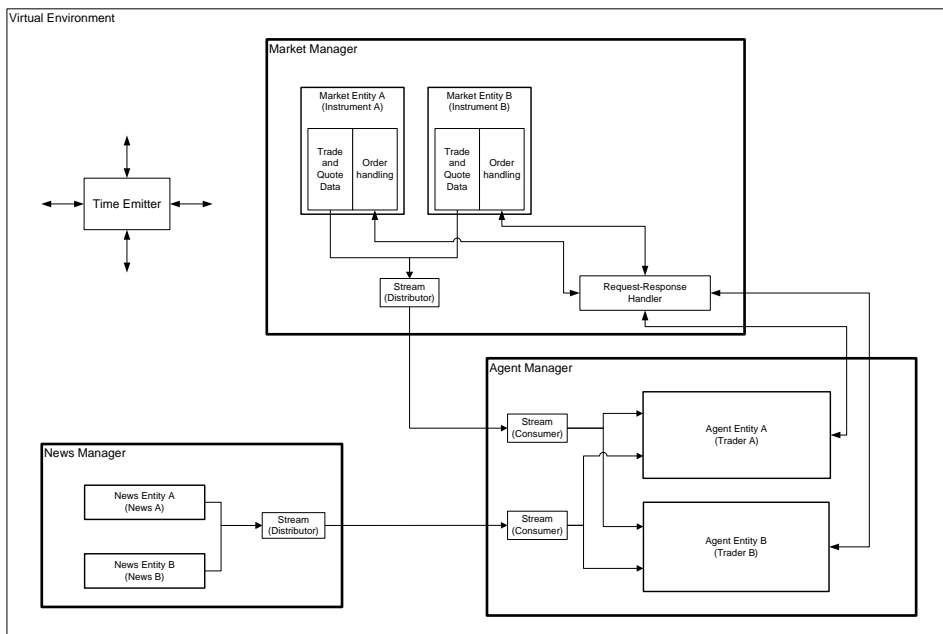


Figure 1: Framework Architecture

The virtual environment provides a Time Emitter that keeps control of virtual time and the communication component that provides the simulation's underlying communication capabilities. The Time Emitter is the central virtual time management component. On the one hand we do not want to have a round based simulation; on the other hand we require the framework to be indifferent towards runtime latencies within the simulation framework. The Time Emitter enables an analysis in a continuous market without simulation framework latencies. It basically is a clock that runs in virtual millisecond intervals which is the usual granularity of stock market trade and quote data. All other components within the simulation framework which need a time signal register at the Time Emitter and receive a timing signal. Simulating a market millisecond by millisecond (even virtual milliseconds) would tremendously increase simulation runtime. In a real stock

market there are often seconds between different quotes or trades. Instead of going from millisecond to millisecond, the Time Emitter requests from all registered components timing information about their next action (e.g. provision of new trade and quote information) or request. The Time Emitter then directly jumps to the closest future timestamp (in ms) of all components and disseminates this timestamp as the new timestamp. This implementation of the Time Emitter greatly reduces simulation runtime.

The communication component is decoupled from the specific stock market simulation which allows for a network transparent extension of the simulation framework. The communication environment offers two different ways of communicating: A one-way stream and a request-response method. Currently the stream is used to disseminate news and market data and the request-response method is used for communication between an agent and the market (order submission information and confirmation). Since the communication environment uses queues (FIFO concept) to transport messages we can include artificially and controlled latencies to study agents' reactions to different system environments. Those latencies can potentially be based on statistical distributions.

The stock market simulation comprises the market and news components as well as agents. Within the market, the Market Manager represents a whole market whereas a Market Entity represents a single instrument. The Market Manager can hold multiple instruments. One of a Market Entity's main functions is the dissemination of trade and quote data for an individual instrument from historical trade and quote data based on the Time Emitter's scheduling. A Market Entity also manages incoming order requests and confirms or denies orders. In our current simulation test runs we model all agents as price takers and the market directly confirms each order. However, our framework is also capable of modeling a competition for liquidity, based on order book information. In such a situation a Market Entity could deny a trade to an agent due to a lack of liquidity.

In the news component, the News Manager comprises all different instrument related news streams (News Entity). It represents an external news or information source. The news component is a passive component that only disseminates news items but does not decide on trades or agents' actions. The News Manager can hold multiple and different news and information sources (News Entity). The agent component is the most important component for a framework user. An agent can access all broadcasted current information, market data and news data. The Agent Manager manages all agents whereas one agent (Agent Entity) resembles one trader or broker in a real stock market. We provide a basic Agent Entity template but users of the simulation framework have complete freedom in implementing additional capabilities into agents as long as they observe the framework's communication interfaces. Users could potentially add libraries for machine-learning or agents with data bases to store trading information and infer decisions from historical data. There are no predefined decision processes within the template agents. The framework provides market and external event information and

users of the framework implement decision rules and systems based on provided information. In our test runs we provide very simple decision rules for testing the stability of the software framework.

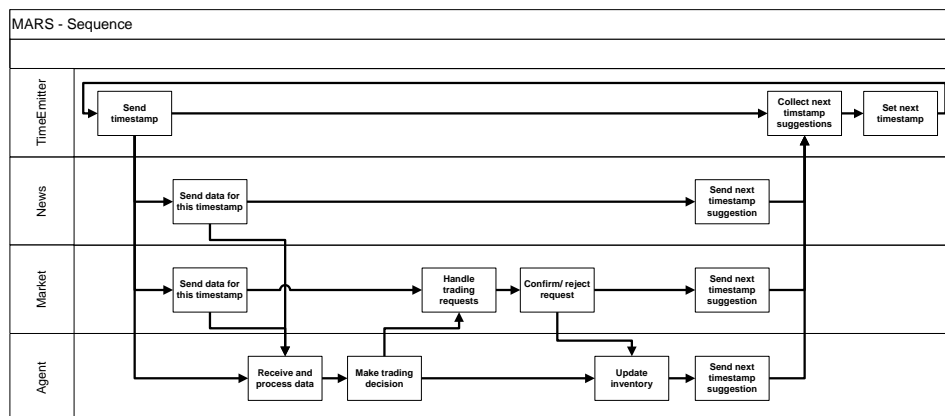


Figure 2: Stylized Sequence Diagram

Figure 2 provides a stylized and simplified sequence diagram. It pictures how different components interact within a simulation run.

With our framework we provide a flexible and extensible platform independent simulation framework for a news or information event and agent based simulation of trading strategies. There is no limit to a user in terms of the interior working of an agent besides computing power and resources.

4 Data and RNSE News

Although our software framework is open to all different types of data sources we use specific data sets to test the simulation framework. Two distinct Thomson Reuters data sources provide input information for the framework. One data stream provides trade and quote information and the second data stream provides pre-quantified news events. It would also be possible to add further parallel data streams to simulate more complicated strategies, e.g. index data that help agents to account for market movements.

The market data, also called trade and quote (TAQ) data, provide the simulation framework with the price process. We obtain these data through the Reuters DataScope Tick History archive³. The data feature an instrument's RIC (Reuters Instrument Code), date, a timestamp to the millisecond, trade price, trade volume, bid, bid size, ask, ask size, and additional qualifiers for special trades and quotes.

³ We thank the Securities Research Centre of Asia-Pacific for providing access to Reuter DataScope Tick History data.

Through TAQ data, the central market system always knows the prevailing bid and ask as well as associated volumes and the last trade with price and volume. Existing qualifiers enable the simulation framework to disregard special trades and quotes that are not in the scope of a specific analysis. Examples are volatility interruptions and intraday auctions on Xetra. TAQ data are more precise than a simple midpoint price process. The availability of bid and ask provides the opportunity to include implicit transactions costs in an analysis. For immediate execution an agent has to pay the quoted half spread if executed at the best bid or ask.

RNSE (Reuters NewsScope Sentiment Engine) data are generated from the normal Thomson Reuters news wire feed (NewsScope). The system uses different techniques and algorithms to score a specific news message with respect to a specific tradable product (Thomson Reuters 2008a, 2008b). The RNSE system generates for each news item, and for each company a news item is relevant for, tags that are either related to *relevance*, *sentiment*, or *novelty*. The scoring is a transformation of ambiguous linguistic content of news wire messages into quantified machine-readable information. The *sentiment* measure gives an indication of the positivity or negativity of news with respect to a certain instrument (company). The *relevance* score provides RNSE users with information to judge the relevance of a news item for a certain instrument. The *novelty* related measures provide an indication whether or not the information in a news item had been published in another news item before.

We use Xetra market and news data for our simulation test run. The simulation runs on the whole year of 2007 based on the 2007 DAX index constituents which were continuously traded. The DAX represents the thirty largest high quality German stocks representing a broad section of industries. TAQ data amount to more than one hundred million data entries and RNSE data to more than 40,000 news messages.

5 Simulation Results of Test Run

To test our simulation we implement various simple agents and run simulation tests based on the data described in the previous section. Since the market in DAX constituents is highly liquid and efficient we realistically do not expect to see positive profits from implemented basic agents but we do expect to see differences between different implementations. However, it is important to note that our main goal here is to test the framework and not to develop trading strategies.

Basically our testing agents react to external news events provided by the RNSE data and based on those data decide on trading with very simple decision rules.

We first implement some basic constraints that agents have to follow: the agents only consider normal trading hours and ignore the first and last five minutes of a trading day to avoid effects due to the opening auction or closing procedure

(9:05 a.m. – 5:25 p.m). Also, volatility interruptions and intraday auctions are not included in the analysis.

Based on RNSE news we implement two different types of agents: one that ignores the *novelty* measures and one that only considers *novel* news (see different results within Table 1 and 2). Agents ignore neutral news items. Basic decision rules for testing require an agent to buy when a news item has a positive *sentiment* and to sell when the *sentiment* is negative. However, agents only consider RNSE news items with a *relevance* measure above 0.5.

Since agents buy and sell based on *sentiment* measures they also have to resolve their inventory based on decision rules to not acquire large portfolios. First of all, we require agents to have no inventory at the end of a trading day. Thus, all agents resolve inventory that exists at the end of a trading day for the then prevailing prices. Throughout the trading day agents only resolve their inventory when they hit a profit barrier. “Hitting a profit” barrier refers to the case that the relative return of a position equals or is greater than a specified value (e.g. 0.01%). Suppose an agent buys 500 shares of Deutsche Telekom for 10.35 EUR and has a profit barrier of 0.01%. If this agent observes a bid price of 10.37 EUR it sells its position in Deutsche Telekom because the relative return is 0.19%. Table 1 shows specific results for 12 agents with different profit barriers.

To include implicit trading costs, agents buy at the best ask and sell at the best bid. Agents usually trade 500 shares, if liquidity (here available quantity) at the best bid or ask is lower they only trade the amount that is available. We provide information about the total loss over the trading year in Euros, information about the per share loss in cents, and a comparison of agents trading only on novel news and agents trading on all news in table 1.

Although we only implement basic agents for testing there is an interesting result. Agents that trade only on novel news consistently have a lower per share loss than agents that trade on all news. One can see that the higher the profit barrier is the lower the losses are. However, with a profit barrier of 3.00% most inventories are only resolved at the end of a trading day.

For our simulation test we use 12 agents on 30 instruments and the whole year of 2007. All data is time stamped to the millisecond. On an Intel QuadCore 2,83 GHz with 8 GB of RAM and Linux operating system, our simulation test run was finished after a little less than 12 hours.

The simulation test run shows that we provide a stable simulation framework with good computational performance. Of course, for further financial analyses more sophisticated trading strategies have to be tested.

Table 1: Overview of test run simulation results

	barrier (%): profit	total loss (in EUR)	per share loss (in ct)	novel news vs. all news (in ct)
novel news	0.01%	50,861	3.9103	1.1919
	0.05%	55,553	4.2727	0.7535
	0.10%	52,961	4.0767	0.7442
	0.50%	35,984	2.7677	1.2886
	1.00%	30,945	2.3803	1.6846
	3.00%	28,090	2.1613	0.2019
all news	0.01%	197,324	5.1023	
	0.05%	194,380	5.0262	
	0.10%	186,366	4.8208	
	0.50%	156,808	4.0562	
	1.00%	157,123	4.0649	
	3.00%	91,346	2.3632	

6 Conclusion

This paper presents a flexible, extensible, and platform independent software framework for an agent and news-event based simulation of algorithmic trading in financial markets. Our simulation model enables researchers and practitioners to simulate time-continuous markets and agents trading on price processes generated from historical data. Additionally, our framework explicitly allows for modeling of exchange system latency. We soundly motivate the usefulness of such a framework in section 2 in the literature review. The architecture of the framework is highly extensible to different data sources and gives a user the freedom to implement an agent with as little constraints as possible. The simulation test run proves the performance and stability of our framework. We believe that our simulation framework is an interesting contribution to support the analysis of news and event based daily and intraday trading strategies.

References

- Antweiler W, Frank MZ (2004). Is all that talk just noise? The information content of internet stock message boards. *Journal of Finance* 59(3): 1139-1167.
- Biais B, Weill PO (2009) Liquidity shocks and order book dynamics. Working Paper. <http://ssrn.com/abstract=1193442>. Last access 2009-08-27.
- Barber BM, Odean T (2008) All that glitters: The effect of attention and news on the buying behavior of individual and institutional investors. *Review of Financial Studies* 21: 785-818.
- Boer K, De Bruin A, Kaymak U (2005) On the design of artificial stock markets. ERIM Report Series Research in Management. Erasmus Universiteit Rotterdam.
- Boer K, Kaymak U, Spiering J (2007) From discrete-time models to continuous-time, asynchronous modeling of financial markets. *Computational Intelligence* 23(2): 142-161.
- Brock WA, Hommes CH (1997) A rational route to randomness. *Econometrica* 65(5): 1059-1095.
- Chaboud A, Chiquoine B, Hjalmarsson E, Vega C (2009) Rise of the machines: Algorithmic trading in the foreign exchange market. Working Paper. http://www.stanford.edu/group/SITE/SITE_2009/segment_1/s1_papers/vvega.pdf. Last access 2009-08-27.
- Copeland TE, Galai D (1983) Information effects on the bid-ask spread. *Journal of Finance* 38(5): 1457-1469.
- Fang L, Peress J (2008) Media coverage and the cross-section of stock returns. Forthcoming in the *Journal of Finance*.
- Green TC (2004) Economic news and the impact of trading on bond prices. *Journal of Finance* 59(3): 1201-1234.
- Hakansson N, Beja A, Kale J (1985) On the feasibility of automated market making by a programmed specialist. *Journal of Finance* 40(1): 1-20.
- Hasbrouck J, Saar G (2009) Technology, latency, strategy. Working Paper.
- Hendershott T, Jones CM, Menkveld AJ (2008) Does algorithmic trading improve liquidity. WFA 2008 Paper. <http://ssrn.com/abstract=1100635>. Last access 2009-08-24.
- Hendershott T, Riordan R (2009) Algorithmic trading and information. Working Paper. Presented at First FBF/IDEI-R Conference on Investment Banking

and Financial Markets, Toulouse. <http://ssrn.com/abstract=1472050>. Last access 2009-09-13.

- Jain P (2005) Financial market design and the equity premium: Electronic versus floor trading. *Journal of Finance* 60(6): 2955-2985.
- Riordan R, Storkenmaier A (2009) Exchange system innovation and algorithmic liquidity supply. Working Paper. Presented at 9th Annual London Business School Transatlantic Doctoral Consortium 2009. <http://ssrn.com/abstract=1247482>. Last access 2009-08-28.
- Shatner M, Mushnik L, Leshno M, Solomon S (2000) A continuous time asynchronous model of the stock market; beyond the LLS model. arXiv:cond-mat/0005430v1. <http://arxiv.org/abs/cond-mat/0005430>. Last access 2009-08-27.
- Stoll HR (2006) Electronic trading in stock markets. *Journal of Economic Perspectives* 20(1): 153-174.
- Tetlock PC (2007) Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance* 62(3): 1139-1167.
- Tetlock PC, Macskassy SA, Saar-Tsechansky M (2008) More than words: Quantifying language to measure firms' fundamentals. *Journal of Finance* 63(3): 1427-1467.
- Tetlock PC (2008) All the news that's fit to reprint: Do investors react to stale information. Columbia University Working Paper. <http://ssrn.com/abstract=1018221>. Last access 2009-09-10.
- Thomson Reuters (2008a) Reuters NewsScope Sentiment Engine: Output image and file format. Thomson Reuters White Paper.
- Thomson Reuters (2008b) Reuters NewsScope Sentiment Engine: Guide to sample data and system overview. Thomson Reuters White Paper.