LivePulse: Tapping Social Media
for Sentiments in Real-Time

Malu Castellanos, Riddhiman Ghosh, Yue Lu, Lei Zhang, Perla Ruiz,
Mohamed Dekhil, Umeshwar Dayal, Meichun Hsu

Hewlett-Packard Laboratories
Palo Alto, CA, USA

{firstname.lastname}@hp.com

ABSTRACT

The rise of Twitter, blogs, review sites and social sites has motivated people to express their opinions publicly and more frequently than ever before. This has fueled the emerging field known as sentiment analysis whose goal is to translate the vagaries of human emotion into hard data. LivePulse is a tool that taps into the growing business interest in what is being said online with the particular characteristic of doing so in real-time. LivePulse integrates novel algorithms for sentiment analysis and a configurable dashboard with different kinds of dynamic charts that change as new data is ingested. It also provides support to drill down and visually explore the sentiment scores to understand how they were computed and what are the emotions expressed about a given aspect or topic. Our tool has been researched and prototyped at HP Labs in close interaction with internal and external customers whose valuable feedback has been crucial for improving the tool. This paper presents an overview of LivePulse's architecture and functionality, and illustrates how it would be demoed.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services.*

General Terms

Algorithms, Design

Keywords

Social media, sentiment analysis, real-time

1. INTRODUCTION

The rapid proliferation of blogs, review sites, Twitter, Facebook, and other Web 2.0 sites, has made it possible for people to publish their opinions more frequently than ever before. The ease with which people can express their thoughts and make them instantaneously available on these sites is a key reason behind this phenomenon. For most businesses, online opinions represent an invaluable source of information. Many such businesses have a team of people dedicated to the task of reading what is posted on

the social and review sites and extracting insight into what is being said about their products and services, or about their competitors products and services. For these businesses opinion analysis has become important for them to remain competitive by listening to their customers (or potential customers) to get actionable insight. However, with the increasing rate at which online opinions are being created, it becomes harder and harder to analyse them by hand and to take immediate, real-time action: for example, reacting to an issue expressed in a blog before its negative opinion spreads and impacts the product sales in the marketplace.

Sentiment analysis is one aspect of opinion mining [1] that tries to capture the polarity of comments and opinions. In other words, it automatically decides if a comment is positive or negative on a topic and may even quantify how positive or negative. For instance, the tweet "The Sorcerer's Apprentice promotion: We have some great news for your tickets...", sentiment analysis needs to identify that it is about a promotion and tickets for the movie and that it is a positive comment about the "promotion" feature related to the movie, and not about the story or the lead actor.

Online sentiment analysis mines the sentiment of comments and opinions from user-generated online content. It provides the ability to automatically monitor such content and mine it in real time to understand the sentiments as they are being expressed and turn them into real-time actionable insight. The importance of this kind of online analysis has motivated recent efforts in developing algorithms and tools for automating the task.

LivePulse is a tool for online sentiment analysis that extracts the polarity of sentiments embedded in online content. It has a number of novel features that are integrated in a single platform distinguishing it from other sentiment analysis tools:

It can deal with noisy data sources like tweets that often have grammatically incorrect English (or other language), use emoticons, colloquial expressions, abbreviations, etc., such as "=D hanging out with fafagraphy friends at bo's it park after watchin sorcerer's, ohhh i liiiked it, hell yeah! ".

It can learn the polarity of opinion words when this polarity is context dependent. For example, in our previous example, the word "hell" which is a negative opinion word in a typical opinion word lexicon, would be placed in a domain-specific positive opinion word lexicon during a previous off-line unsupervised learning phase. As another example of a context dependent opinion word, the word "large" is a positive word for the size of a laptop screen but it is a negative word for the size of the battery.

It can identify the polarity of non-standard English words such as the word "liiike" (which stands for "like") in the above example.

It analyzes not only the overall sentiment of a text but also the sentiment about different aspects that it discovers about a topic.

It can handle real-time streams by streaming data to in-memory data structures where all the computations are performed without requiring any access to the database.

It features charts that dynamically change (some actually move) as data streams in and is analysed, to show how the sentiment on a set of topics is evolving over time.

It allows the user to visually explore the sentiment scores to easily understand how they were computed while at the same time getting insight into the emotions expressed about a given aspect or topic.

It has a configurable dashboard that lets the user specify the streaming data source, the topic(s) to monitor and optionally other parameters such as the time window size to display on the charts, the refresh rate and the aggregation period (e.g., aggregate the sentiment of the last hour) if the default values are not suitable. It also enables the user to pause the sentiment monitoring session and save it to replay (on the dynamic charts) and explore it later.

LivePulse is a research project that started at HP Labs. It has been used internally by several business units to get insight into what people are saying about HP products, what they like about HP's products and what they are complaining about. Throughout the development of the tool, we maintained very close interaction with internal and a few external customers to create a tool that would meet their needs. It is important to note that LivePulse applies to stored data as well as real-time streaming data.

In this paper we give an overview of the architecture of LivePulse and its main components including a hint on the algorithms (section 2). We then present a scenario to describe the demo, followed by conclusions and future work.

2. OVERVIEW AND ARCHITECTURE

In this section we provide a brief overview of the LivePulse system architecture, as shown in Figure 1. LivePulse is a cloud service that provides REST web service interfaces. It is architected to operate over real-time streaming data sources (such as Twitter), frequently updated content sources (such as review sites), and historical / stored content (such as previously crawled data).

Content ingestion adapters pull data from instances of these different source types, which have differing schema and characteristics, into LivePulse. We also allow for plug-in adapters to accommodate new data sources. Data obtained as a result of the content ingestion is fed to the Sentiment Processor. The constituent modules of the Sentiment Processor, as shown in Figure 1, are described below, along with a brief overview of the algorithms underlying their operation.

Pre-Processor and Data Cleanser: This module is responsible for the pre-processing and cleansing required to be performed on collected data to make it amenable for analysis by further stages of the LivePulse pipeline. In case of tweets, it removes spam tweets (they only introduce noise) and retweets (duplicates that skew analysis results), and restores popular abbreviations to their corresponding original forms using a lexicon of abbreviations (e.g., "wknd" to "weekend"). Other content types may require different kinds of cleansing.

NLP Tasks: This module performs several natural language processing tasks required by the other stages of the LivePulse pipeline, including decomposing text into sentences, splitting sentences into appropriate tokens, and tagging them with their part-of-speech. We use the open-source OpenNLP package to perform these tasks.

Attribute Extractor: Opinion is expressed not just on entities, but at a finer granularity on attributes of entities. For example, the sentence, "The picture of the TV is awesome", expresses a positive opinion on the picture attribute of the TV entity. Discovering the attributes of entities mentioned in text such as tweets is an important step in the Sentiment Process. To extract attributes we use the recently proposed double propagation technique [2] that is a domain independent and unsupervised technique. It is based on the observation that "opinion words" (word that express opinion or contain sentiment) are usually associated with attributes in the corpus. Thus, opinion words can be recognized by identified attributes, and attributes can be identified by known opinion words.

We use a modified version of a publicly available lexicon of opinion words. This propagation process stops when no more opinion words or attributes can be found from the corpus, leaving behind a set of discovered attributes.

Attribute Clustering: Since the techniques described in this paper compute attribute-specific sentiment polarity rather than overall sentiment, it is especially important in LivePulse to be able navigate, interpret and consume the extracted attributes effectively. However attribute extraction techniques, including those describe above, tend to produce a large number of attributes

Figure 1: LivePulse System Architecture
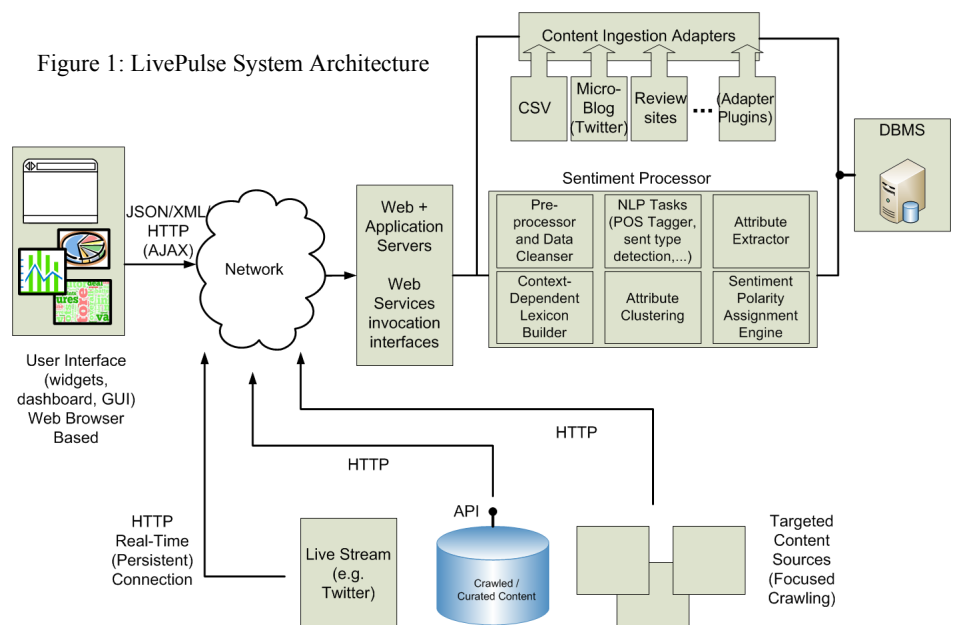
from corpuses of even a modest size. While simply sorting the list of discovered attributes by their frequency of occurrence to determine their relative importance is one way of managing the discovered attributes, it leaves a lot to be desired as it does not account for the "long-tail" of emergent attributes effectively.

We employ a number of techniques to first clean, normalize and then cluster the discovered attributes into semantically cohesive categories by using unsupervised machine learning. Since most of the corpuses we are using have been sourced from the Web, the emergent attributes have been observed to be noisy, replete with misspellings, and variations in morphology.

We employ heuristics to resolve observed misspellings based on Levenshtein distance, and then perform lemmatization to normalize morphological variations. We have developed clustering algorithms that use the WordNet lexical database to compute semantic distance between attributes, based on their relative distances in the WordNet hypernym/hyponym tree, to cluster the attributes into cohesive categories. To deal with domain-specific attributes, which are not found in the WordNet lexicon, we are currently experimenting with community-curated knowledge bases such as FreeBase, which typically have much larger coverage. We also allow for domain-expert tuning of the automatically generated attribute hierarchy.

Sentiment Polarity Assignment Engine: The next important step, after identifying attributes, is to assign a sentiment value for those attributes, based on the opinion words that apply to them.

The relations between opinion words and attributes are identified via dependency relations using an approximate dependency parser. (For computational efficiency reasons, we do not use a full fledged dependency parser, but rather use pattern-based heuristics to approximate dependency parsing). For instance, in "The phone has an amazing screen", the relation between the opinion word "amazing" and the attribute "screen" is detected through the mod (modifier) dependency that connects them. In [2], two categories of dependency relations, direct and indirect relations, are defined to summarize all types of dependency relations between two words. Both categories are useful but for tweets only the direct category is used since the indirect one is error-prone.

Once the relation between opinion words and attributes is determined, polarity of sentiment (positive, negative or neutral) has to be assigned to the attributes. We use an opinion lexicon-based approach to assign polarity, where domain-specific opinion lexicons are used to identify particular opinion words, and based on the relation identified sentiment polarity can be determined for the attribute directly. If no clear dependency relation exists either the opinion word closest to the attribute or the opinion words within a window of a given size are used to compute the attribute polarity. Negation words (e.g. "not", "n't", "never") are considered to reverse polarity within their scope.

Comparative sentences, of the form "the X is much better than Y", are dealt with specially, since neither dependency grammar nor closest opinion word rules may apply [3]. First the presence of a comparative word is detected by its corresponding POS tag, such as JJR (comparative adjective), RBR (comparative verb), JJS(superlative adjective) and RBS(superlative adverb). From these tags, the superior/inferior entities are found. Based on English grammar, if the sentence is positive, then the entities before the comparative keyword are superior; otherwise they are inferior. The effects of negations are considered here too.

Context-Dependent Lexicon Builder: A well known challenge in lexicon-based sentiment analysis approaches, such as ours, is the sensitivity of the polarity of a sentiment word to the topic domain. For example, "unpredictable" is negative in the domain of an electronic appliance, but positive in the movie domain. Moreover, even in the same topic domain, the same word may indicate different polarities with respect to different aspects of the topic. For example, "large" is negative for waiting line but positive for audience. Aiming at solving this challenge, we have developed a novel approach [4] to construct context-dependent sentiments automatically.

Our method does not require labeled content which is very expensive to obtain in a new topic domain. Instead, we look for several kinds of signals that are helpful in determining the context-dependent sentiments: a) General sentiment lexicon, which contains words that are almost always positive or negative in any domain, such as "excellent" and "bad". This lexicon provides high confidence but low coverage sentiment words. b) Synonym/antonym dictionary, such as in WordNet. For example, we may not know "large" is positive or negative for laptop screen, but we know it should be very similar to "big" and very different than "tiny". c) Other sentiment signals available in the corpus, which convey information about the sentiment expressed in the text. For tweets, emoticons convey this information. d) Language heuristics. We can often infer the polarity relation of phrases based on some language patterns. For example, "aspect 1 is A *but* aspect 2 is B" usually indicates that word A and B are of the opposite polarity. By the same token, "aspect 1 is A *and* aspect 2 is B" typically indicates the same polarity for words A and B.

To solve the challenge that multiple signals come in different format and may even cause contradictions, we combine them through appropriate constraints in the objective function of a novel optimization framework [4]. Basically we search for optimal assignments of sentiment scores to aspect-opinion pairs that are most consistent with all the constraints. We have obtained a convex objective function that makes it possible for the optimal solutions to be found using convex optimization algorithms. In this way, we have a principled and efficient method to combine all the useful signals and construct the context-dependent sentiment automatically.

This is an off-line task whose goal is to incrementally build domain-dependent and context-aware sentiment lexicons. We expect this lexicon to grow more rapidly for any new domain initially, and then reach a point where the additions begin to taper off.

3. DEMO DESCRIPTION

The purpose of the demo is to show the main features of LivePulse:
a) the "live" aspect enabling it to listen to online content, analyze it and display the results, all in real time;
b) its "pulsing" capability to sense the feeling of what is being said on the web;
c) its dynamic visualization features to get insight into the instantaneous evolution of the monitored topics;
d) its exploratory capability that enables the user to understand how the sentiment scores were computed;

For the demo we will have LivePulse listening to tweets on a few topics that are hot at the time of the WWW 2011 conference. We will show how LivePulse enables the user to get different levels of

insight about sentiments. Starting from sentiments about topics and continuing to sentiments about aspects or attributes of the topics, to end with the specific emotions expressed in the tweets and how they influence the sentiment scores.

As tweets start arriving they are displayed on the dashboard and the charts start to fill up and dynamically move as new tweets arrive, as illustrated in Figure 2. We plan to ask the spectator to send a few positive and negative tweets on those topics (either from a smart phone or from the browser of another laptop that we will have there) so that they experience how as their tweets appear on the LivePulse's dashboard right then and there, the charts change accordingly. The line chart (upper chart in figure 2) will be shifting to the right as new points appear every 10 seconds. This will show how the average sentiment about the topic of the corresponding line is changing. The bar charts (bottom charts in figure 2, one chart per topic) will also shift to the right as new bars come to show the evolution in the proportion of positive, negative and neutral tweets about the corresponding topic. We will also show a tag cloud visualization (Figure 3) of the tweets that changes dynamically. The tag cloud allows to get a quick summary of the importance (in terms of frequency reflected in the tag size) of the different aspects or attributes (those discovered by the tool) of the monitored topics and their average sentiment (reflected in the color and color gradient). When the user will click on a given tag, a pie chart showing the proportion of positive, negative and neutral sentiments about that attribute will appear. Then the spectator will be allowed to do further exploration to get more insight about the attribute sentiments. In particular, when he will click on an attribute (i.e., tag), a table with all the tweets and their sentiment scores (Figure 4) will appear displaying the sentiment oriented words color-coded according to their influence on the score. The tweet text is also visually marked to indicate the scope of negations. Finally, when the spectator will click on a given tweet, the sentiment scoring formula appears first instantiated with the sentiment oriented words and their degree of influence and then instantiated with the numeric values.
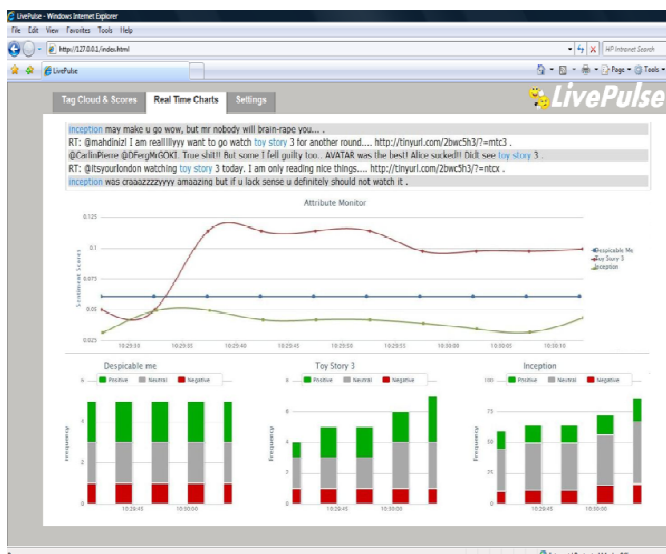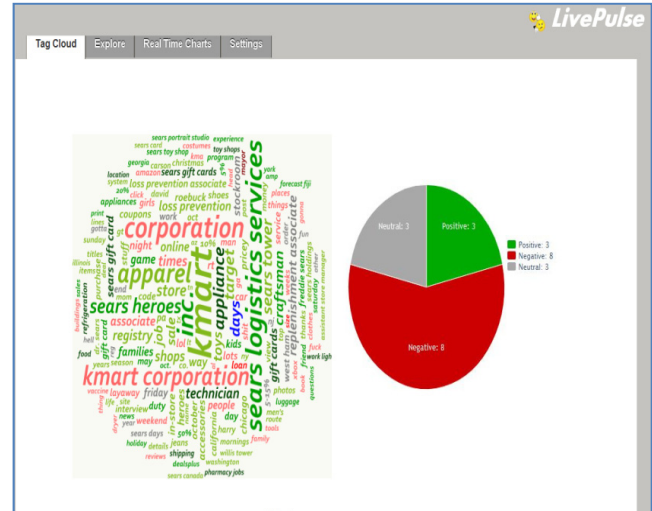

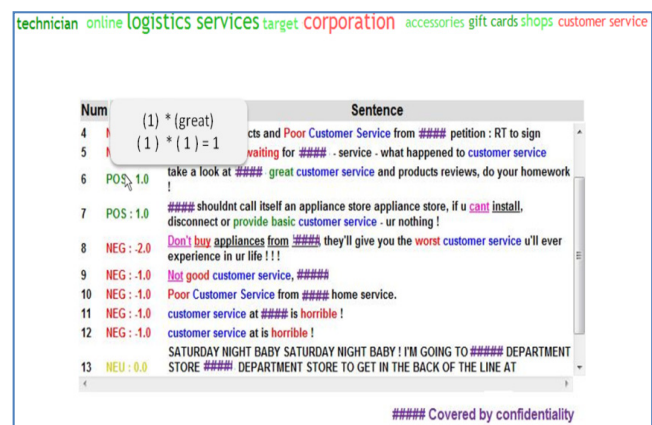Figure 3. Tag Cloud and rationale

Figure 4: Exploring sentiment scores

4. CONCLUSIONS

LivePulse is a sentiment analysis tool that has been successfully used to understand customers' opinions published in social media about HP's products . The demo will show its real-time capabilities as applied to Twitters, and its ability to effectively extract the sentiment of incoming tweets and display the shifting sentiments dynamically over time.

5. REFERENCES

[1] Pang, Bo and Lee, Lillian. Opinion Mining and Sentiment Analysis. Foundations and Trends In Information Retrieval. Vol 2. Nos. 1-2 (2008), 1-135.

[2] Qiu, Guang., Bing, Liu., Jiajun Bu and Chun Chen. Expanding Domain Sentiment Lexicon through Double Propagation. In Proceedings of IJCAI 2009.

[3] Ding, Xiaowen, Bing Liu and Lei Zhang. Entity discovery and assignment for opinion mining applications.In Proceedings of KDD 2009.

[4] Lu, Yue, Malu Castellanos, Umesh Dayal and ChengXiang Zhai. Automatic Construction of a Context-Aware Sentiment Lexicon: An Optimization Approach. Submitted to WWW 2011.

Figure 2: Dynamic charts in LivePulse