Structured Models for Fine-to-Coarse Sentiment Analysis

R. McDonald K. Hannan T. Neylon M. Wells J. Reynar

Google, Inc.
Presented by Abraham Erdos

January 20, 2011

Outline

Motivation

What is sentiment analysis?

- Sentiment analysis aims to determine the attiutde of a writer or speaker on some topic.
- Multi-level sentiment analysis is important.
- Different needs for different applications.

Motivation

Different Needs for Different Applications

Product Review Summarization

- Sentence or phrase level polarity classification.

Question Ansering System

- Paragraph level sentiment classification

Document Type Analysis

- Document level sentiment classification

Motivation
Example

> *This is the first Mp3 player that I have used ... I thought it sounded great ... After only a few weeks, it started having trouble with the earphone connection ...* ***I won't be buying another.***
>
> *Mp3 player review from Amazon.com*

- Negative review with positive and negative sentences.

Motivation
Example

- Why not make a separat esystem for each level of granularity?

> *My 11 year old daughter has also been using it and it is a lot harder than it looks.*

- Positive review with a single negative sentence.
- Fitness Equipment: Hard $\rightarrow$ Good Workout
- "Hard" sentiment can only be determined in context.

Structured Model

- $Y(d)$: discrete set of sentiment labels at document level $\{pos, neg\}$
- $Y(s)$: discrete set of sentiment labels at sentence level e.g., $\{pos, neu, neg\}$
- Input document contains sentences $\mathbf{s} = s_1, \ldots, s_n$
  - Must produce sentiment labels for the document, $y^d \in Y(d)$
- Each sentence is labeled $\mathbf{y}^s = y_1^s, \ldots, y_n^s$, where $y_i^s \in Y(s)\ \forall\ 1 \leq i \leq n$

Structured Model

- Deinfe $\mathbf{y}$ as the joint labeling of the document and sentences $\mathbf{y} = (y^d, \mathbf{y}^s) = (y^d, y_1^s, \ldots, y_n^s)$

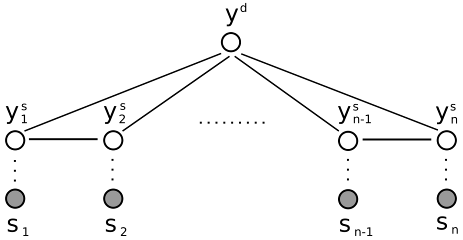

Figure: Sentence and document level model.

Structured Model

- Structured linear classifiers [Collins(2002)] are used to score the document

$$\begin{aligned} score\left(\mathbf{y},\mathbf{s}\right) &= score\left(\left(y^{d},\mathbf{y}^{s}\right),\mathbf{s}\right) \\ &= score\left(\left(y^{d},y_{1}^{s},\ldots,y_{n}^{s}\right),\mathbf{s}\right) \\ &= \sum_{i=2}^{n} score\left(\left(y^{d},y_{i-1}^{s},\ldots,y_{i}^{s}\right),\mathbf{s}\right) \end{aligned}$$

Inference as Sequential Labeling

- Each clique score is a linear combination of features and their weights.

$$score\left(\left(y^d, y^s_{i-1}, \ldots, y^s_i\right), \mathbf{s}\right) = \mathbf{w} \cdot \mathbf{f}\left(\left(y^d, y^s_{i-1}, \ldots, y^s_i\right), \mathbf{s}\right)$$

- $\mathbf{f}$ is a high dimensional feature representation fo the clique.
- $\mathbf{w}$ is a corresponding weight vector.
- $\mathbf{s}$ is the input vector of sentences.

Inference as Sequential Labeling

- The inference problem is to find the higest scoreing for the labeling $\mathbf{y}$ for an input $\mathbf{s}$

$$\arg\max_{\mathbf{y}} \{score(\mathbf{y}, \mathbf{s})\}$$

- If the label y^d is fixed, then inference in the model from Figure 1 reduces to the sequential case.
- Search space is only over the sentence labels y_i^s, whose graphical structure forms a chain.
- Given y^d, the sentiment labels for $\mathbf{s}$ can be solved using the Viterbi algorithm

Inference as Sequential Labeling

Input: $\mathbf{s} = s_1, \ldots, s_n$

1. $\mathbf{y} = null$
2. for each $y^d \in Y(d)$
3. $\quad \mathbf{y}^s = \arg\max_{y^s} score\left(\left(y^d, \mathbf{y}^s\right), \mathbf{s}\right)$
4. $\quad \mathbf{y}^{'} = \left(y^d, \mathbf{y}^s\right)$
5. $\quad$ if $score\left(\mathbf{y}^{'}, \mathbf{s}\right) > score(\mathbf{y}, \mathbf{s})$ or $\mathbf{y} = null$
6. $\qquad \mathbf{y} = \mathbf{y}^{'}$
7. return $\mathbf{y}$

- Line 3 is solved using the Viterbi algorithm for a fixed y^d.

Feature Space

- Feature space for each clique is $\mathbf{f}\left(\left(y^d, y_{i-1}^s, \ldots, y_i^s\right), \mathbf{s}\right)$.
- Each sentence s_i is represented by a set of binary predicates $P(s_i)$.
  - Can contain any predicate.
  - Here includes all unigram, bigram, and trigrams in s_i.
  - Obtained using an automatic classifier.

Example predicates in $P(s)$

a:DT_great:JJ_product:NN
a:DT_great:JJ_*:NN
a:DT_*:JJ_product:NN
*:DT_great:JJ_product:NN
a:DT_*:JJ_*:NN

Feature Space

- Each predicate, p, is conjoined with the label information to contruct a binary feature.

Example Features

- Given $Y(s) = \{subj, obj\}$ and $Y(d) = \{pos, neg\}$

$$\mathbf{f}_{(j)}\left(\left(y^d, y^s_{i-1}, \ldots, y^s_i\right), \mathbf{s}\right) = \begin{cases} 1 & \text{if } p \in P(s_i) \\ & \text{and } y^s_{i-1} = obj \\ & \text{and } y^s_i = subj \\ & \text{and } y^d = neg \\ 0 & \text{otherwise} \end{cases}$$

Training

- Weights $\mathbf{w}$ are trained using the MIRA learning algorithm [Crammer and Singer(2003)].
    - Inference based online large-margin learning technique.
    - Relies only on inference to learn the weight vector.
    - Has been shown to provide state-fo-the-art accuracy.

Training

Training data: $T = \{(\mathbf{y}_t, \mathbf{s}_t)\}_{t=1}^{T}$

1. $\mathbf{w}^{(0)} = 0; \; i = 0$
2. for $n : 1..N$
3. for $t : 1..N$
4. $\mathbf{w}^{(i+1)} = \arg\min_{\mathbf{w}^*} \left\| \mathbf{w}^* - \mathbf{w}^{(i)} \right\|$
   s.t. $score\,(\mathbf{y}_t, \mathbf{s}_t) - score\left(\mathbf{y}^{'}, \mathbf{s}\right) \geq L\left(\mathbf{y}_t, \mathbf{y}^{'}\right)$ relative
   to $\mathbf{w}^* \forall \mathbf{y}^{'} \in \mathscr{C} \subset Y$, where $|\mathscr{C}| = k$
5. return $\mathbf{w}^{(N \times T)}$

- Weight **w** is updated in line 4 through quadratic programming.

Beyond Two-Level Models

- Longer documents may benefit from document, paragraph, and sentence level analysis.

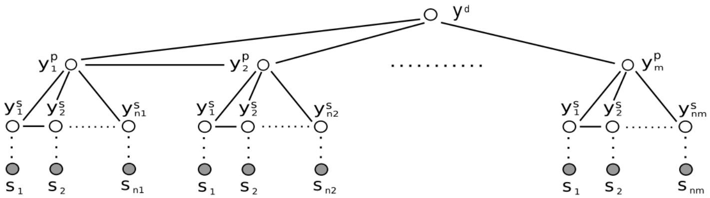

Figure: An extension to the model from Figure 1 incorporates paragraph level analysis.

Experiments

- Corpus of 600 online product reviews
  - Duplicates discarded.
  - Insufficent text discarded.
  - Spam discarded.
- Three different typse of products reviewed
  - Car seats for children.
  - Fitness equipment.
  - MP3 players.

Experiments

- Reviews labeled by online customers $Y(d) = \{pos, neg\}$
- Every sentence anotated $Y(s) = \{pos, neu, neg\}$.

	Sentence Stats				Document Stats		
	Pos	Neg	Neu	Tot	Pos	Neg	Tot
Car	472	443	264	1179	98	80	178
Fit	568	635	371	1574	92	97	189
MP3	485	464	214	1163	98	89	187
Tot	1525	1542	849	3916	288	266	554

Experiments

- Three baseline systems were created

Document-Classifier is a classifier that learns to predict the document label only.

Sentence-Classifier is a classifier that leans to predict sentence labels in isolation of one another, i.e., without recard for the document or neighbor sentences.

Sentence-Structured is a sentence classifier that uses a sequential chain model to learn and classify sentences. It is essentially Figure 1 without the top level document node.

Results

	Sentence Accuracy				Document Accuracy			
	Car	Fit	MP3	Total	Car	Fit	MP3	Total
Doc	-	-	-	-	72.8	80.1	**87.2**	80.3
Sent	54.8	56.8	49.4	53.1	-	-	-	-
Sent-Str	60.5	61.4	55.7	58.8	-	-	-	-
Jo-Str	**63.5**	**65.2**	**60.1**	**62.6**	**81.5**	**81.9**	85.0	**82.8**

St→Dc	60.5	61.4	55.7	58.8	75.9	80.7	86.1	81.1
Dc→St	59.7	61.0	58.3	59.5	72.8	80.1	87.2	80.3

Summary

- Investigated the use of a globa structured model that learns to predict sentiment on different levels of granularity.
- Experiments show that this model obtains higher accuracy than classifiers trained in isolation as well as cascaded sytems that pass information from one level to another at test time.

- Further Work
  - Extend models for partially labeled data
  - Use of relative positions of phrases and cues, e.g., "in conclusion" or "to summarize"

For Further Reading I

Michael Collins.
Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms.
In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
doi: http://dx.doi.org/10.3115/1118693.1118694.
URL `http://dx.doi.org/10.3115/1118693.1118694`.

For Further Reading II

Koby Crammer and Yoram Singer.
Ultraconservative online algorithms for multiclass problems.
J. Mach. Learn. Res., 3:951–991, March 2003.
ISSN 1532-4435.
doi: http://dx.doi.org/10.1162/jmlr.2003.3.4-5.951.
URL `http://dx.doi.org/10.1162/jmlr.2003.3.4-5.951`.