

Embarrassingly Simple Unsupervised Aspect Extraction

Stéphan Tulkens

CLiPS

University of Antwerp

Belgium

stephan.tulkens@uantwerpen.be

Andreas van Cranenburgh

Department of Information Science

University of Groningen

The Netherlands

a.w.van.cranenburgh@rug.nl

Abstract

We present a simple but effective method for aspect identification in sentiment analysis. Our unsupervised method only requires word embeddings and a POS tagger, and is therefore straightforward to apply to new domains and languages. We introduce Contrastive Attention (CA_t 🐱), a novel single-head attention mechanism based on an RBF kernel, which gives a considerable boost in performance and makes the model interpretable. Previous work relied on syntactic features and complex neural models. We show that given the simplicity of current benchmark datasets for aspect extraction, such complex models are not needed. The code to reproduce the experiments reported in this paper is available at <https://github.com/clips/cat>.

1 Introduction

We consider the task of unsupervised aspect extraction from text. In sentiment analysis, an aspect can intuitively be defined as a dimension on which an entity is evaluated (see Figure 1). While aspects can be concrete (e.g., a laptop battery), they can also be subjective (e.g., the loudness of a motorcycle). Aspect extraction is an important subtask of aspect-based sentiment analysis. However, most existing systems are supervised (for an overview, cf. Zhang et al., 2018). As aspects are domain-specific, supervised systems that rely on strictly lexical cues to differentiate between aspects are unlikely to transfer well between different domains (Rietzler et al., 2019). Another reason to consider the unsupervised extraction of aspect terms is the scarcity of training data for many domains (e.g., books), and, more importantly, the complete lack of training data for many languages. Unsupervised aspect extraction has previously been attempted with topic models (Mukherjee and Liu, 2012), topic model hybrids (García-Pablos et al., 2018), and re-

The two things that really drew me to *vinyl* were the **expense** and the **inconvenience**.

Figure 1: An example of a sentence expressing two aspects (red) on a target (*italics*). Source: <https://www.newyorker.com/cartoon/a19180>

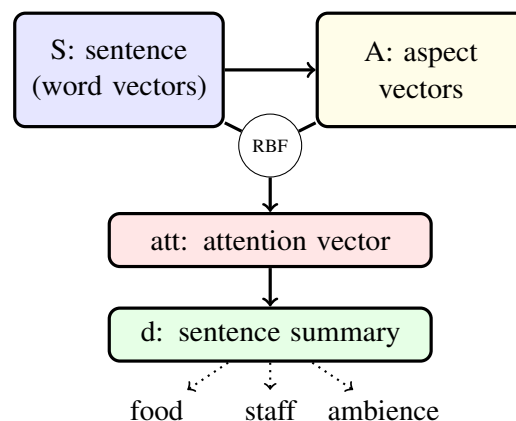


Figure 2: An overview of our aspect extraction model.

stricted Boltzmann machines (Wang et al., 2015), among others. Recently, autoencoders using attention mechanisms (He et al., 2017; Luo et al., 2019) have also been proposed as a method for aspect extraction, and have reached state of the art performance on a variety of datasets. These models are unsupervised in the sense that they do not require labeled data, although they do rely on unlabeled data to learn relevant patterns. In addition, these are complex neural models with a large number of parameters. We show that a much simpler model suffices for this task.

We present a simple unsupervised method for aspect extraction which only requires a POS tagger and in-domain word embeddings, trained on a small set of documents. We introduce a novel single-head attention mechanism, Contrastive At-

the bread is top notch as well .
 best spicy tuna roll , great asian salad .
 also get the onion rings – best we ’ve ever had .

Figure 3: Examples of Contrastive Attention ($\gamma=0.03$)

tention (CA_t 🍷), based on Radial Basis Function (RBF) kernels. Compared to conventional attention mechanisms (Weston et al., 2014; Sukhbaatar et al., 2015), CA_t 🍷 captures more relevant information from a sentence. Our method outperforms more complex methods, e.g., attention-based neural networks (He et al., 2017; Luo et al., 2019). In addition, our method automatically assigns aspect labels, while in previous work, labels are manually assigned to aspect clusters. Finally, we present an analysis of the limitations of our model, and propose some directions for future research.

2 Method

Like previous methods (Hu and Liu, 2004; Xu et al., 2013), our method (see Figure 2) consists of two steps: extraction of candidate aspect terms and assigning aspect labels to instances. Both steps assume a set of in-domain word embeddings, which we train using `word2vec` (Mikolov et al., 2013). We use a small set of in-domain documents, containing about 4 million tokens for the restaurant domain.

Step 1: aspect term extraction In previous work (Hu and Liu, 2004; Xu et al., 2013), the main assumption has been that nouns that are frequently modified by sentiment-bearing adjectives (e.g., good, bad, ugly) are likely to be aspect nouns. We experimented with this notion and devised a labeling strategy in which aspects are extracted based on their co-occurrence with seed adjectives. However, during experimentation we found that for the datasets in this paper, the most frequent nouns were already good aspects; any further constraint led to far worse performance on the development set. This means that our method only needs a POS tagger to recognize nouns, not a full-fledged parser. Throughout this paper, we use `spaCy` (Honnibal and Montani, 2017) for tokenization and POS tagging. In Section 5, we investigate how these choices impact performance.

Step 2: aspect selection using Contrastive Attention We use a simple form of attention, similar to the attention mechanism used in memory

networks (Weston et al., 2014; Sukhbaatar et al., 2015). With an attention mechanism, a sequence of words, e.g., a sentence or a document, is embedded into a matrix S , which is operated on with an aspect a to produce a probability distribution, att . Schematically:

$$\text{att} = \text{softmax}(aS) \quad (1)$$

att is then multiplied with S to produce an informative summary with respect to the aspect a :

$$d = \sum_i \text{att}_i S_i \quad (2)$$

Where d is the weighted sentence summary. There is no reason to restrict a to be a single vector: when replaced by a matrix of queries, A , the equation above gives a separate attention distribution for each aspect, which can then be used to create different summaries, thereby keeping track of different pieces of information. In our specific case, however, we are interested in tracking which words elicit aspects, regardless of the aspect to which they belong. We address this by introducing Contrastive Attention (CA_t 🍷), a way of calculating attention that integrates a set of query vectors into a single attention distribution. It uses an RBF kernel, which is defined as follows:

$$\text{rbf}(x, y, \gamma) = \exp(-\gamma \|x - y\|_2^2) \quad (3)$$

where, x and y are vectors, and γ is a scaling factor, which we treat as a hyperparameter. An important aspect of the RBF kernel is that it turns an arbitrary unbounded distance, the squared euclidean distance in this case, into a bounded similarity. For example, regardless of γ , if x and y have a distance of 0, their RBF response will be 1. As their distance increases, their similarity decreases, and will eventually asymptote towards 0, depending on γ . Given the RBF kernel, a matrix S , and a set of aspect vectors A , attention is calculated as follows:

$$\text{att} = \frac{\sum_{a \in A} \text{rbf}(w, a, \gamma)}{\sum_{w \in S} \sum_{a \in A} \text{rbf}(w, a, \gamma)} \quad (4)$$

The attention for a given word is thus the sum of the RBF responses of all vectors in A , divided by the sum of the RBF responses of the vectors to all vectors in S . This defines a probability distribution over words in the sentence or document, where words that are, on average, more similar to aspects, get assigned a higher score.

	Train	Test
Citysearch (2009)		1,490
SemEval (2014)	3,041	402
SemEval (2015)	1,315	250

Table 1: The number of sentences in each of the datasets after removing sentences that did not express exactly one aspect in our set of aspects.

Method	P	R	F
SERBM (2015)	86.0	74.6	79.5
ABAE (2017)	89.4	73.0	79.6
W2VLDA (2018)	80.8	70.0	75.8
AE-CSA (2019)	85.6	86.0	85.8
Mean	78.9	76.9	77.2
Attention	80.5	80.7	80.6
CAt 🍷	86.5	86.4	86.4

Table 2: Weighted macro averages across all aspects on the test set of the Citysearch dataset.

Step 3: assigning aspect labels After reweighing the word vectors, we label each document based on the cosine similarity between the weighted document vector d and the label vector.

$$\hat{y} = \operatorname{argmax}_{c \in C} (\cos(d, \vec{c})) \quad (5)$$

Where C is the set of labels, i.e., {FOOD, AMBIENCE, STAFF}. In the current work, we use word embeddings of the labels as the targets. This avoids the inherent subjectivity of manually assigning aspect labels, the strategy employed in previous work (He et al., 2017; Luo et al., 2019).

3 Datasets

We use several English datasets of restaurant reviews for the aspect extraction task. All datasets have been annotated with one or more sentence-level labels, indicating the aspect expressed in that sentence (e.g., the sentence “The sushi was great” would be assigned the label FOOD). We evaluate our approach on the Citysearch dataset (Ganu et al., 2009), which uses the same labels as the SemEval datasets. To avoid optimizing for a single corpus, we use the restaurant subsets of the SemEval 2014 (Pontiki et al., 2014) and SemEval 2015 (Pontiki et al., 2015) datasets as development data. Note that, even though our method is completely unsupervised, we explicitly allocate test data to ensure proper methodological soundness,

Method	P	R	F
Aspect: FOOD			
SERBM (2015)	89.1	85.4	87.2
ABAE (2017)	95.3	74.1	82.8
W2VLDA (2018)	96.0	69.0	81.0
AE-CSA (2019)	90.3	92.6	91.4
Mean	92.4	73.5	85.6
Attention	86.7	89.5	88.1
CAt 🍷	91.8	92.4	92.1

Aspect: STAFF			
SERBM (2015)	81.9	58.2	68.0
ABAE (2017)	80.2	72.8	75.7
W2VLDA (2018)	61.0	86.0	71.0
AE-CSA (2019)	92.6	75.6	77.3
Mean	55.8	85.7	67.5
Attention	74.4	69.3	71.8
CAt 🍷	82.4	75.6	78.8

Aspect: AMBIENCE			
SERBM (2015)	80.5	59.2	68.2
ABAE (2017)	81.5	69.8	74.0
W2VLDA (2018)	55.0	75.0	64.0
AE-CSA (2019)	91.4	77.9	77.0
Mean	58.7	56.1	57.4
Attention	67.1	65.7	66.4
CAt 🍷	76.6	80.1	76.6

Table 3: Precision, recall, and F-scores on the test set of the Citysearch dataset.

and do not optimize any models on the test set. Following previous work (He et al., 2017; Ganu et al., 2009), we restrict ourselves to sentences that only express exactly one aspect; sentences that express more than one aspect, or no aspect at all, are discarded. Additionally, we restrict ourselves to three labels: FOOD, SERVICE, and AMBIENCE. We adopt these restrictions in order to compare to other systems. Additionally, previous work (Brody and Elhadad, 2010) reported that the other labels, ANECDOTES and PRICE, were not reliably annotated. Table 1 shows statistics of the datasets.

4 Evaluation

We optimize all our models on SemEval ’14 and ’15 training data; the scores on the Citysearch dataset do not reflect any form of optimization with regards to performance. We optimize the hyperparameters of each model separately (i.e., the number of aspect terms and γ of the RBF kernel), leading to the following hyperparameters: For the regular

attention, we select the top 980 nouns as aspect candidates. For the RBF attention, we use the top 200 nouns and a γ of .03.

We compare our system to four other systems. W2VLDA (García-Pablos et al., 2018) is a topic modeling approach that biases word-aspect associations by computing the similarity from a word to a set of aspect terms. SERBM (Wang et al., 2015) a restricted Boltzmann Machine (RBM) that learns topic distributions, and assigns individual words to these distributions. In doing so, it learns to assign words to aspects. We also compare our system to two attention-based systems. First, ABAE (He et al., 2017), which is an auto-encoder that learns an attention distribution over words in the sentence by simultaneously considering the global context and aspect vectors. In doing so, ABAE learns an attention distribution, as well as appropriate aspect vectors. Second, AE-CSA (Luo et al., 2019), which is a hierarchical model which is similar to ABAE. In addition to word vectors and aspect vectors, this model also considers sense and sememe (Bloomfield, 1926) vectors in computing the attention distribution. Note that all these systems, although being unsupervised, do require training data, and need to be fit to a specific domain. Hence, all these systems rely on the existence of in-domain training data on which to learn reconstructions and/or topic distributions. Furthermore, much like our approach, ABAE, AE-CSA, and W2VLDA rely on the availability of pre-trained word embeddings. Additionally, AE-CSA needs a dictionary of senses and sememes, which might not be available for all languages or domains. Compared to other systems, our system does require a UD POS tagger to extract frequent nouns. However, this can be an off-the-shelf POS tagger, since it does not need to be trained on domain-specific data.

We also compare our system to a baseline based on the mean of word embeddings, a version of our system using regular attention, and a version of our system using Contrastive Attention (CA_t). The results are shown in Table 3. Because of class imbalance (60 % of instances are labeled FOOD), the F-scores from 3 do not give a representative picture of model performance. Therefore, we also report weighted macro-averaged scores in Table 2.

Our system outperforms ABAE, AE-CSA, and the other systems, both in weighted macro-average F1 score, and on the individual aspects. In addition, 2 shows that the difference between ABAE and

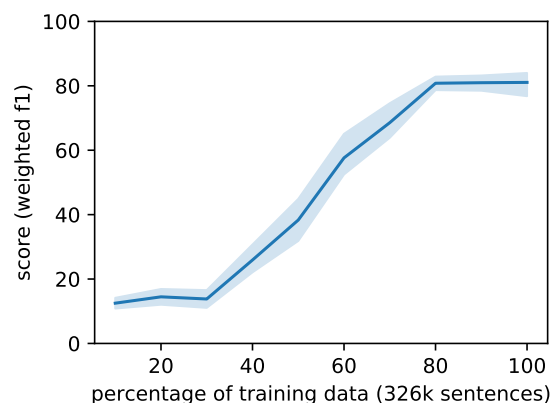


Figure 4: A learning curve on the restaurant data, averaged over 5 embedding models.

SERBM is smaller than one would expect based on the F1 scores on the labels, on which ABAE outperforms SERBM on STAFF and AMBIENCE. The Mean model still performs well on this dataset, while it does not use any attention or knowledge of aspects. This implies that aspect knowledge is probably not required to perform well on this dataset; focusing on lexical semantics is enough.

5 Analysis

We perform an ablation study to see the influence of each component of our system; specifically, we look at the effect of POS tagging, in-domain word embeddings, and the amount of data on performance.

Only selecting the most frequent words as aspects, regardless of their POS tag, had a detrimental effect on performance, giving an F-score of 64.5 (Δ -21.9), while selecting nouns based on adjective-noun co-occurrence had a smaller detrimental effect, giving an F-score of 84.4 (Δ -2.2), higher than ABAE and SERBM.

Replacing the in-domain word embeddings trained on the training set with pretrained GloVe embeddings (Pennington et al., 2014)¹ had a large detrimental effect on performance, dropping the F-score to 54.4 (Δ -32); this shows that in-domain data is important.

To investigate how much in-domain data is required to achieve good performance, we perform a learning curve experiment (Figure 4). We increase the training data in 10% increments, training five word2vec models at each increment. As the fig-

¹Specifically, the glove.6B.200D vectors from <https://nlp.stanford.edu/projects/glove/>

Phenomenon	Example
OOV	“I like the Somosas”
Data Sparsity	“great Dhal”
Homonymy	“Of course”
Verb > Noun	“Waited for food”
Discourse	“She didn’t offer dessert”
Implicature	“No free drink”

Table 4: A categorization of observed error types.

ure shows, only a modest amount of data (about 260k sentences) is needed to tackle this specific dataset.

To further investigate the limits of our model, we perform a simple error analysis on our best performing model. Table 4 shows a manual categorization of error types. Several of the errors relate to Out-of-Vocabulary (OOV) or low frequency items, such as the words ‘Somosas’ (OOV) and ‘Dhal’ (low frequency). Since our model is purely based on lexical similarity, homonyms and polysemous words can lead to errors. An example of this is the word ‘course,’ which our model interprets as being about food. As the aspect terms we use are restricted to nouns, the model also misses aspects expressed in verbs, such as “waited for food.” Finally, discourse context and implicatures often lead to errors. The model does not capture enough context or world knowledge to infer that ‘no free drink’ does not express an opinion about drinks, but about service.



Given these errors, we surmise that our model will perform less well in domains in which aspects are expressed in a less overt way. For example, consider the following sentence from a book review (Kirkus Reviews, 2019):

- (1) As usual, Beaton conceals any number of surprises behind her trademark wry humor.

This sentence touches on a range of aspects, including writing style, plot, and a general opinion on the book that is being reviewed. Such domains might also require the use of more sophisticated aspect term extraction methods.

However, it is not the case that our model necessarily overlooks implicit aspects. For example, the word “cheap” often signals an opinion about the price of something. As the embedding of the word “cheap” is highly similar to that of “price” our model will attend to “cheap” as long as enough price-related terms are in the set of extracted aspect terms of the model.

In the future, we would like to address the limitations of the current method, and apply it to datasets with other domains and languages. Such datasets exist, but we have not yet evaluated our system on them due to the lack of sufficient unannotated in-domain data in addition to annotated data.

Given the performance of CAAt , especially compared to regular dot-product attention, it would be interesting to see how it performs as a replacement of regular attention in supervised models, e.g., memory networks (Weston et al., 2014; Sukhbaatar et al., 2015). Additionally, it would be interesting to see why the attention model outperforms regular dot product attention. Currently, our understanding is that the dot-product attention places a high emphasis on words with a higher vector norm; words with a higher norm have, on average, a higher inner product with other vectors. As the norm of a word embedding directly relates to the frequency of this word in the training corpus, the regular dot-product attention naturally attends to more frequent words. In a network with trainable parameters, such as ABAE (He et al., 2017), this effect can be mitigated by finetuning the embeddings or other weighting mechanisms. In our system, no such training is available, which can explain the suitability of CAAt  as an unsupervised aspect extraction mechanism.

6 Conclusion

We present a simple model of aspect extraction that uses a frequency threshold for candidate selection together with a novel attention mechanism based on RBF kernels, together with an automated aspect assignment method. We show that for the task of assigning aspects to sentences in the restaurant domain, the RBF kernel attention mechanism outperforms a regular attention mechanism, as well as more complex models based on auto-encoders and topic models.

Acknowledgments

We are grateful to the three reviewers for their feedback. The first author was sponsored by a Fonds Wetenschappelijk Onderzoek (FWO) aspirantschap.

References

Leonard Bloomfield. 1926. A set of postulates for the science of language. *Language*, 2(3):153–164.

- Samuel Brody and Noemie Elhadad. 2010. [An unsupervised aspect-sentiment model for online reviews](#). In *Proceedings of NAACL-HLT*, pages 804–812.
- Gayatree Ganu, Noemie Elhadad, and Amélie Marian. 2009. [Beyond the stars: improving rating predictions using review text content](#). In *Proceedings of WebDB*, volume 9, pages 1–6.
- Aitor García-Pablos, Montse Cuadros, and German Rigau. 2018. [W2VLDA: almost unsupervised system for aspect based sentiment analysis](#). *Expert Systems with Applications*, 91:127–137.
- Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. [An unsupervised neural attention model for aspect extraction](#). In *Proceedings of ACL*, pages 388–397.
- Matthew Honnibal and Ines Montani. 2017. [spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing](#). Software package.
- Minqing Hu and Bing Liu. 2004. [Mining and summarizing customer reviews](#). In *Proceedings of ACM SIGKDD*, pages 168–177.
- Kirkus Reviews. 2019. [Beating about the bush](#).
- Ling Luo, Xiang Ao, Yan Song, Jinyao Li, Xiaopeng Yang, Qing He, and Dong Yu. 2019. [Unsupervised neural aspect extraction with sememes](#). In *Proceedings of IJCAI*, pages 5123–5129.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). In *ICLR Workshop Papers*.
- Arjun Mukherjee and Bing Liu. 2012. [Aspect extraction through semi-supervised modeling](#). In *Proceedings of ACL*, pages 339–348.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of EMNLP*, pages 1532–1543.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. [Semeval-2015 task 12: Aspect based sentiment analysis](#). In *Proceedings of SemEval*, pages 486–495.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. [Semeval-2014 task 4: Aspect based sentiment analysis](#). In *Proceedings of SemEval*.
- Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2019. [Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification](#). arXiv preprint arXiv:1908.11860.
- Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. [End-to-end memory networks](#). In *Proceedings of NIPS*, pages 2440–2448.
- Linlin Wang, Kang Liu, Zhu Cao, Jun Zhao, and Gerard de Melo. 2015. [Sentiment-aspect extraction based on restricted boltzmann machines](#). In *Proceedings of ACL-IJCNLP*, pages 616–625.
- Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. [Memory networks](#). arXiv preprint arXiv:1410.3916.
- Liheng Xu, Kang Liu, Siwei Lai, Yubo Chen, and Jun Zhao. 2013. [Mining opinion words and opinion targets in a two-stage framework](#). In *Proceedings of ACL*, pages 1764–1773.
- Lei Zhang, Shuai Wang, and Bing Liu. 2018. [Deep learning for sentiment analysis: A survey](#). *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.