An Unsupervised Approach to Product Attribute Extraction

Santosh Raju, Prasad Pingali, and Vasudeva Varma

Language Technologies Research Center,
IIIT Hyderabad, India
santosh_raju@research.iiit.ac.in,{pvvpr,vv}@iiit.ac.in

Abstract. Product Attribute Extraction is the task of automatically discovering attributes of products from text descriptions. In this paper, we propose a new approach which is both unsupervised and domain independent to extract the attributes. With our approach, we are able to achieve 92% precision and 62% recall in our experiments. Our experiments with varying dataset sizes show the robustness of our algorithm. We also show that even a minimum of 5 descriptions provide enough information to identify attributes.

1 Introduction

Automatic information extraction from product descriptions can greatly reduce human effort in manually identifying attributes and represent the product features in a structured manner. This would also be useful in many applications such as product recommendation systems. In this paper, we focus on extraction of attributes from a set of input documents that describe a particular product.

Our system takes a set of text documents which are descriptions of varieties of the same product as input and outputs a list of attributes specific to that product. These documents contain a list of sentences which are often phrases and incomplete sentences. Some samples include *'Widescreen LCD Monitor'*, *'1.3 mega pixel camera to capture those special moments'* and *'Modified QWERTY keyboard with SureType predictive text technology for easy messaging'*.

There has been some recent work on product attributes extraction. The system presented in [1] extracts attribute-value pairs from text descriptions using semi-supervised learning techniques. They extract attribute-value pair instances from product descriptions for a particular domain whereas our system ouputs a list of attributes for a particular product. In our unsupervised approach, we extract attributes by clustering noun phrases whereas [1] treats the extraction task as a word classification problem. Other past approaches [2, 3] include systems which extract product features from product reviews.

We develop our algorithm following the intuition that an attribute should occur multiple times in different product descriptions. Our unsupervised approach has three steps: Pre-processing, Clustering and Attribute Extraction. In pre-processing, relevant noun phrases are identified from the input text documents as explained in section 2. The details of clustering are given in section 3, where we group similar noun phrases. Ideally, a cluster should contain all the

phrases that describe a particular attribute and different clusters should describe different attributes. Section 4 describes the method we used for extracting an attribute from each cluster. We then discuss our experimental results in section 5 and finally conclude in section 6 showing interesting future directions.

2 Pre-Processing

The goal of this step is to process the text documents and identify noun phrases which are given as input to clustering. Sentences from these documents are tagged with parts of speech using Brill's Tagger and noun phrases are extracted.

Product descriptions contain phrases which begin with a determiner word like "your favorite music", "those special moments" and other single word noun phrases like "comfort", "reliability" which often explain an attribute of the product rather than define it.

We employ two pruning methods to eliminate the above noun phrases. In the first method, we simply discard all the noun phrases which begin with a determiner word. In the second method, we assume that single word noun phrases mentioned above occur more frequently in general English than in product descriptions. Let p and q be the unigram probability distributions of input document set and a general English corpus respectively. Now we compute pointwise KL divergence(used in [4]) score δ_w for each unigram w in the input documents which gives the relative importance of the unigram in the input document set compared to the generic corpus.

$$\delta_w(p||q) = p(w)log\frac{p(w)}{q(w)} \tag{1}$$

Now we eliminate all the single word noun phrases whose δ_w is less than a threshold θ. We define θ as the sum of mean(μ) and standard deviation(σ) of the δ values of all unigrams w in the input document set.

We have observed that pruning the noun phrases in this manner doesn't effect the recall of the system because most of the attribute instances which are wrongly pruned at this stage have occurred in other noun phrases.

3 Clustering

The noun phrases obtained from the previous step are clustered so that noun phrases describing the same attribute are grouped together in the same cluster. Thus each cluster represents occurrences of a particular attribute.

3.1 Similarity Measure

We calculate N gram overlap to measure the similarity between two noun phrases. We consider unigram and bigram overlap for this. Bigrams are ordered pairs of words co-occurring within five words of each other. Let S_i and S_j be the sets of uni-grams, bi-grams belonging to two noun phrases P_i and P_j respectively. Now

we define the similarity between the two noun phrases P_i and P_j using Dice's Coefficient similarity: $Sim(P_i, P_j) = \frac{2|S_i \cap S_j|}{|S_i|+|S_j|}$. We prepare a similarity matrix containing pairwise noun phrase similarities which is used by the clustering algorithm.

3.2 Noun Phrase Clustering

We use Group Average Agglomerative Clustering(GAAC) algorithm for computing the clusters. GAAC treats each input datapoint as a separate singleton cluster and then iteratively merges the most similar cluster-pair in each step. Here, the similarity between two clusters is equal to the average similarity between their datapoints. We continue merging until the average distance between the datapoints of the clusters being merged is less than α times the maximal distance between them where $\alpha < 1$ is constant. α value is set to 0.6 in our experiments.

3.3 Pruning

The output clusters thus obtained contain clusters of varying size. The chance of finding an attribute in a cluster increases with the size of the cluster. In our experiments, we consider clusters that contain at least three noun phrases and discard all clusters containing one or two noun phrases. This simple pruning method employed on the clusters makes sense as it conforms with our assumption that attribute instances occur multiple times in different input documents.

Before extracting the attributes from the clusters, we remove the generic units of measure from noun phrases. A list containing 40 units of measure(*cm*,*kg* etc) is prepared for this purpose which is given as input to the system.

4 Attribute Extraction

Assuming that each cluster has noun phrases that contain instances of same attribute, an attribute is extracted from each cluster. We use the following techniques to extract attribute name from a cluster.

We compute unigrams, bigrams and trigrams from the noun phrases in a cluster and these N-grams together form the candidates for attribute name. The problem now boils down to selecting the best N-gram. We define an Attribute Scoring Function AR to score each of these N-grams. We declare that the N-gram with highest score is the attribute. The Attribute Score of an N-gram x is computed as follows,

$$AS(x) = \frac{PKL(x)}{AHD(x)} \tag{2}$$

Where PKL is the pointwise KL divergence score and AHD is the Average Head Noun Distance of the N-gram. Let P be the probability distribution of this cluster and Q be the probability distribution of the rest of the clusters together. Then the PKL score of an N-gram x is $PKL(x) = P(x)log\frac{P(x)}{Q(x)}$

A high PKL score is reached by high frequency in this cluster and low frequency in other clusters whereas a low PKL score is reached by low frequency in this cluster and high frequency in other clusters. Thus PKL score gives the importance of an N-gram to a particular cluster.

We define AHD as the average head noun distance of the N-gram x in its instances. Head Noun Distance is the distance of the N-gram x from the right most word(head noun) in the noun phrase with the minimum distance being 1. For example, in the noun phrase "Widescreen LCD Monitor", the Head Noun Distance D of the N-grams "Widescreen LCD", "LCD Monitor", "Monitor" are 2, 1 and 1 respectively.

$$AHD(x) = \frac{\Sigma_i D(x,i)}{N(x)} \tag{3}$$

Where $D(x,i)$ is the head noun distance of ith instance of N-gram x and $N(x)$ is the number of instances of N-gram x in the cluster.

The intuition behind the AHD metric is that in most of the cases where a value-attribute pair appears in a phrase, value is written first followed by attribute even if they are not just single words but N-grams. So, an N-gram close to the head noun of the phrase has more chance of being an attribute when compared to N-grams with larger AHD values.

Thus an N-gram with high PKL score and low AHD score should be selected as an attribute. So we define the Attribute Score as the ratio of PKL score and AHD score. We find attributes from all the clusters by using the Attribute Ranking Function and its performance is evaluated as explained in section 5.

5 Evaluation

Data We carried out experiments on 4 datasets containing product descriptions collected from www.amazon.com website. Each dataset has 25 descriptions each describing a different variety of the same product. The products we considered for our experiments are "Cell Phone", "Microwave Oven", "Watch" and "Portable DVD Player". We have manually prepared list of all the attributes found in the text documents for each of the 4 products. The performance of our system is evaluated by matching the output attributes with the manually extracted attributes.

Precision and Recall We measure the precision of the system in two ways: fully correct, partially correct attributes similar to the method used in [1]. A match is fully correct if the system extracts an attribute that completely matches with an attribute in the manually prepared list. If the system extracts an attribute that partially matches with an attribute in the manually prepared list, then the attribute is partially correct attribute. For example, an attribute "Monitor" from manually prepared list will partially match an attribute "LCD Monitor" extracted by the system. We consider an attribute recalled if it is partially

or fully correct. Precision and Recall on different datasets are presented in table 1.

A preliminary analysis of the system's output showed that most of the attributes extracted are actually important attributes of that product which is not entirely surprising as our approach is based on the word frequencies in the descriptions.

Effect of Dataset Size Table 1 gives the average precision and average recall against varying number of descriptions in each dataset. Our approach is able to get 90% precision even for a small input dataset of 5 documents, though a compromise in recall is observed. Recall improves with increasing dataset size as it provides more evidence for identifying attributes. The robustness of the approach is seen with consistent precision, irrespective of dataset size.

	25 docs					15 docs	5 docs
	DVD	Cellphone	Watch	Oven	Average	Average	Average
Precision (Fully Correct)	47.5	45.2	57.1	62.1	**51.5**	**57.6**	**64.5**
Precision (Fully or Partially Correct)	92.6	88.1	100	93.1	**92.4**	**92.3**	**90.3**
Recall	72.0	63.1	55.5	57.8	**62.7**	**45.8**	**28.6**

Table 1. Precision and Recall (in percentages)

6 Conclusion

In this paper, we presented a new unsupervised and domain independent approach for attribute extraction from product descriptions which yielded promising results. We proved the robustness of our system by presenting results for datasets of different sizes. We used the notion of *'clustering the noun phrases'* to group attribute instances to extract the attributes. As a future work, we plan to use similar notion to solve other related problems such as "Synonymy Identification in Attributes", "Hierarchical Attribute Extraction" which require cluster comparisons once the attributes are extracted.

References

1. Probst, K., Ghani, R., Krema, M., Fano, A.E., Liu, Y.: Semi-supervised learning of attribute-value pairs from product descriptions. In: IJCAI. (2007)
2. Scaffidi, C., Bierhoff, K., Chang, E., Felker, M., Ng, H., Jin, C.: Red opal: product-feature scoring from reviews. In: EC '07: Proceedings of the 8th ACM conference on Electronic commerce, ACM (2007)
3. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: HLT '05: Proceedings of the conference on HLT and EMNLP, ACL (2005)
4. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 workshop on Multiword expressions, ACL (2003)