Unsupervised Disambiguation of Image Captions

Wesley May, Sanja Fidler, Afsaneh Fazly, Sven Dickinson, and Suzanne Stevenson
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada, M5S 3G4
{wesley,fidler,afsaneh,sven,suzanne}@cs.toronto.edu

Abstract

Given a set of images with related captions, our goal is to show how visual features can improve the accuracy of unsupervised word sense disambiguation when the textual context is very small, as this sort of data is common in news and social media. We extend previous work in unsupervised text-only disambiguation with methods that integrate text and images. We construct a corpus by using Amazon Mechanical Turk to caption sense-tagged images gathered from ImageNet. Using a Yarowsky-inspired algorithm, we show that gains can be made over text-only disambiguation, as well as multimodal approaches such as Latent Dirichlet Allocation.

1 Introduction

We examine the problem of performing unsupervised word sense disambiguation (WSD) in situations with little text, but where additional information is available in the form of an image. Such situations include captioned newswire photos, and pictures in social media where the textual context is often no larger than a tweet.

Unsupervised WSD has been shown to work very well when the target word is embedded in a large quantity of text (Yarowsky, 1995). However, if the only available text is “The crane was so massive it blocked the sun” (see Fig. 1), then text-only disambiguation becomes much more difficult; a human could do little more than guess. But if an image is available, the intended sense is much clearer. We develop an unsupervised WSD algorithm based on Yarowsky’s that uses words in a short caption along with “visual words” from the captioned image to choose the best of two possible senses of an ambiguous keyword describing the content of the image.

Language-vision integration is a quickly developing field, and a number of researchers have explored the possibility of combining text and visual features in various multimodal tasks. Leong and Mihalcea (2011) explored semantic relatedness between words and images to better exploit multimodal content. Jamieson et al. (2009) and Feng and Lapata (2010) combined text and vision to perform effective image annotation. Barnard and colleagues (2003; 2005) showed that supervised WSD by could be improved with visual features. Here we show that unsupervised WSD can similarly be improved. Loeff, Alm and Forsyth (2006) and Saenko and Darrell (2008) combined visual and textual information to solve a related task, image sense disambiguation, in
an unsupervised fashion. In Loeff et al.’s work, little
gain was realized when visual features were added
to a great deal of text. We show that these features
have more utility with small textual contexts, and
that, when little text is available, our method is more
suitable than Saenko and Darrell’s.

2 Our Algorithm

We model our algorithm after Yarowsky’s (1995) al-
gorithm for unsupervised WSD: Given a set of doc-
uments that contain a certain ambiguous word, the
goal is to label each instance of that word as some
particular sense. A seed set of collocations that
strongly indicate one of the senses is initially used to
label a subset of the data. Yarowsky then finds new
collocations in the labelled data that are strongly as-
associated with one of the current labels and applies
these to unlabelled data. This process repeats iter-
atively, building a decision list of collocations that
indicate a particular sense with a certain confidence.

In our algorithm (Algorithm 1), we have a docu-
ment collection $D$ of images relevant to an ambigu-
ous keyword $k$ with senses $s_1$ and $s_2$ (though the
algorithm is extensible to more than two senses). Such
a collection might result from an internet image
search using an ambiguous word such as “mouse”.

Each $D_i$ is an image–caption pair repesented as a
bag-of-words that includes both lexical words from
the caption, and “visual words” from the image. A
visual word is simply an abstract representation that
describes a small portion of an image, such that simi-
lar portions in other images are represented by the
same visual word (see Section 3.2 for details). Our
seed sets consist of the words in the definitions of $s_1$
and $s_2$ from WordNet (Fellbaum, 1998). Any document
whose caption contains more words from one
sense definition than the other is initially labelled
with that sense. We then iterate between two steps
that (i) find additional words associated with $s_1$
or $s_2$ in currently labelled data, and (ii) relabel all data
using the word sense associations discovered so far.

We let $V$ be the entire vocabulary of words across
all documents. We run experiements both with and
without visual words, but when we use visual words,
they are included in $V$. In the first step, we com-
pute a confidence $C_i$ for each word $V_i$. This con-
fidence is a log-ratio of the probability of seeing
$V_i$ in documents labelled as $s_1$ as opposed to doc-
uments labelled as $s_2$. That is, a positive $C_i$ indi-
cates greater association with $s_1$, and vice versa. In
the second step we find, for each document $D_j$, the
word $V_i \in D_j$ with the highest magnitude of $C_i$. If
the magnitude of $C_i$ is above a labelling threshold
$\tau_c$, then we label this document as $s_1$ or $s_2$
depending on the sign of $C_i$. Note that all old labels are
discarded before this step, so labelled documents may
become unlabelled, or even differently labelled, as
the algorithm progresses.

Algorithm 1 Proposed Algorithm

| $D$: set of documents $D_1 \ldots D_d$ |
| $V$: set of lexical and visual words $V_1 \ldots V_v$ in $D$ |
| $C_i$: log-confidence $V_i$ is sense 1 vs. sense 2 |
| $S_1$ and $S_2$: bag of dictionary words for each sense |
| $L_1$ and $L_2$: documents labelled as sense 1 or 2 |

for all $D_i$ do
$\triangleright$ Initial labelling using seed set
if $|D_i \cap S_1| > |D_i \cap S_2|$ then
$L_1 \leftarrow L_1 \cup \{D_i\}$
else if $|D_i \cap S_1| < |D_i \cap S_2|$ then
$L_2 \leftarrow L_2 \cup \{D_i\}$
end if
end for

repeat

for all $i \in 1..v$ do
$\triangleright$ Update word conf.
$C_i \leftarrow \log \left( \frac{P(V_i | L_1)}{P(V_i | L_2)} \right)$
end for

$L_1 \leftarrow \emptyset$, $L_2 \leftarrow \emptyset$ $\triangleright$ Update document conf.

for all $D_i$ do
$\triangleright$ Find word with highest confidence
$m \leftarrow \arg \max_{j \in 1..v, V_j \in D_i} |C_j|$
if $C_m > \tau_c$ then
$L_1 \leftarrow L_1 \cup \{D_i\}$
else if $C_m < -\tau_c$ then
$L_2 \leftarrow L_2 \cup \{D_i\}$
end if
end for

until no change to $L_1$ or $L_2$

3 Creation of the Dataset

We require a collection of images with associated
captions. We also require sense annotations for
the keyword for each image to use for evaluation.
Barnard and Johnson (2005) developed the
3.1 Captioning Images

To develop a large set of sense-annotated image-caption pairs with a focus on caption-sized text, we turned to ImageNet (Deng et al., 2009). ImageNet is a database of images that are each associated with a synset from WordNet. Hundreds of images are available for each of a number of senses of a wide variety of common nouns. To gather captions, we used Amazon Mechanical Turk to collect five sentences for each image. We chose two word senses for each of 20 polysemous nouns and for each sense we collected captions for 50 representative images. For each image we gathered five captions, for a total of 10,000 captions. As we have five captions for each image, we split our data into five sets. Each set has the same images, but each image is paired with a different caption in each set.

We specified to the Turkers that the sentences should be relevant to, but should not talk directly about, the image, as in “In this picture there is a blue fish”, as such captions are very unnatural. True captions generally offer orthogonal information that is not readily apparent from the image. The keyword for each image (as specified by ImageNet) was not presented to the Turkers, so the captions do not necessarily contain it. Knowledge of the keyword is presumed to be available to the algorithm in the form of an image tag, or filename, or the like. We found that forcing a certain word to be included in the caption also led to sentences that described the picture very directly. Sentences were required to be at least ten words long, and have acceptable grammar and spelling. We remove stop words from the captions and lemmatize the remaining words. See Figure 2 for some examples.

3.2 Computing the Visual Words

We compute visual words for each image with ImageNet’s feature extractor. This extractor lays down a grid of overlapping squares onto the image and computes a SIFT descriptor (Lowe, 2004) for each square. Each descriptor is a vector that encodes the edge orientation information in a given square. The descriptors are computed at three scales: 1x, 0.5x and 0.25x the original side lengths. These vectors are clustered with k-means into 1000 clusters, and the labels of these clusters (arbitrary integers from 1 to 1000) serve as our visual words.

It is common for each image to have a “vocabulary” of over 300 distinct visual words, many of which only occur once. To denoise the visual data, we use only those visual words which account for at least 1% of the total visual words for that image.

4 Experiments and Results

To show that the addition of visual features improves the accuracy of sense disambiguation for image-caption pairs, we run our algorithm both with and without the visual features. We also compare our results to three different baseline methods: K-means (K-M), Latent Dirichlet Allocation (LDA) (Blei et al., 2003), and an unsupervised WSD algorithm (PBP) explained below. We use accuracy to measure performance as it is commonly used by the WSD community (See Table 1).

For K-means, we set $k = 2$ as we have two senses, and represent each document with a $V$-dimensional
vector, where the i-th element is the proportion of word $V_i$ in the document. We run K-means both with and without visual features.

For LDA, we use the dictionary sense model from Saenko and Darrell (2008). A topic model is learned where the relatedness of a topic to a sense is based on the probabilities of that topic generating the seed words from its dictionary definitions. Analogously to k-means, we learn a model for text alone, and a model for text augmented with visual information.

For unsupervised WSD (applied to text only), we use WordNet::SenseRelate::TargetWord, hereafter PBP (Patwardhan et al., 2007), the highest scoring unsupervised lexical sample word sense disambiguation algorithm at SemEval07 (Pradhan et al., 2007). PBP treats the nearby words around the target word as a bag, and uses the WordNet hierarchy to assign a similarity score between the possible senses of words in the context, and possible senses of the target word. As our captions are fairly short, we use the entire caption as context.

The most important result is the gain in accuracy after adding visual features. While the average gain across all words is slight, it is significant at $p < 0.02$ (using a paired t-test). For 12 of the 20 words, the visual features improve performance, and in 6 of those, the improvement is 5–11%.

For some words there is no significant improvement in accuracy, or even a slight decrease. With words like “bass” or “chip” there is little room to improve upon the text-only result. For words like “plant” or “press” it seems the text-only result is not strong enough to help bootstrap the visual features in any useful way. In other cases where little improvement is seen, the problem may lie with high intra-class variation, as our visual words are not very robust features, or with a lack of orthogonality between the lexical and visual information.

Our algorithm also performs significantly better than the baseline measurements. K-means performs surprisingly well compared to the other baselines, but seems unable to make much sense of the visual information present. Saenko and Darrell’s (2008) LDA model makes substantial gains by using visual features, but does not perform as well on this task. We suspect that a strict adherence to the seed words may be to blame: while both this LDA model and our algorithm use the same seed definitions initially, our algorithm is free to change its mind about the usefulness of the words in the definitions as it progresses, whereas the LDA model has no such capacity. Indeed, words that are intuitively non-discriminative, such as “carry”, “lack”, or “late”, are not uncommon in the definitions we use.

<table>
<thead>
<tr>
<th>Word</th>
<th>Ours text w/vis</th>
<th>Ours text</th>
<th>K-M text w/vis</th>
<th>K-M text</th>
<th>LDA text w/vis</th>
<th>LDA text</th>
<th>PBP text w/vis</th>
<th>PBP text</th>
</tr>
</thead>
<tbody>
<tr>
<td>bank</td>
<td>.77</td>
<td>.78</td>
<td>.66</td>
<td>.65</td>
<td>.64</td>
<td>.56</td>
<td>.62</td>
<td>.49</td>
</tr>
<tr>
<td>band</td>
<td>.90</td>
<td>.90</td>
<td>.90</td>
<td>.88</td>
<td>.61</td>
<td>.62</td>
<td>.49</td>
<td>.57</td>
</tr>
<tr>
<td>crane</td>
<td>.62</td>
<td>.67</td>
<td>.76</td>
<td>.76</td>
<td>.52</td>
<td>.54</td>
<td>.66</td>
<td>.55</td>
</tr>
<tr>
<td>game</td>
<td>.78</td>
<td>.78</td>
<td>.60</td>
<td>.66</td>
<td>.60</td>
<td>.66</td>
<td>.70</td>
<td>.55</td>
</tr>
<tr>
<td>hood</td>
<td>.74</td>
<td>.73</td>
<td>.73</td>
<td>.70</td>
<td>.51</td>
<td>.45</td>
<td>.55</td>
<td>.50</td>
</tr>
<tr>
<td>jack</td>
<td>.76</td>
<td>.74</td>
<td>.62</td>
<td>.53</td>
<td>.58</td>
<td>.66</td>
<td>.47</td>
<td>.50</td>
</tr>
<tr>
<td>mold</td>
<td>.81</td>
<td>.92</td>
<td>.79</td>
<td>.54</td>
<td>.57</td>
<td>.70</td>
<td>.50</td>
<td>.50</td>
</tr>
<tr>
<td>mouse</td>
<td>.84</td>
<td>.84</td>
<td>.71</td>
<td>.62</td>
<td>.62</td>
<td>.69</td>
<td>.68</td>
<td>.68</td>
</tr>
<tr>
<td>plant</td>
<td>.54</td>
<td>.54</td>
<td>.56</td>
<td>.53</td>
<td>.52</td>
<td>.50</td>
<td>.72</td>
<td>.70</td>
</tr>
<tr>
<td>press</td>
<td>.60</td>
<td>.59</td>
<td>.60</td>
<td>.54</td>
<td>.58</td>
<td>.62</td>
<td>.48</td>
<td>.48</td>
</tr>
<tr>
<td>seal</td>
<td>.70</td>
<td>.80</td>
<td>.61</td>
<td>.67</td>
<td>.55</td>
<td>.53</td>
<td>.62</td>
<td>.62</td>
</tr>
<tr>
<td>speaker</td>
<td>.70</td>
<td>.69</td>
<td>.57</td>
<td>.53</td>
<td>.55</td>
<td>.62</td>
<td>.63</td>
<td>.63</td>
</tr>
<tr>
<td>squash</td>
<td>.89</td>
<td>.95</td>
<td>.84</td>
<td>.92</td>
<td>.55</td>
<td>.67</td>
<td>.79</td>
<td>.79</td>
</tr>
<tr>
<td>track</td>
<td>.78</td>
<td>.85</td>
<td>.71</td>
<td>.66</td>
<td>.51</td>
<td>.54</td>
<td>.69</td>
<td>.69</td>
</tr>
<tr>
<td>avg.</td>
<td>.76</td>
<td>.78</td>
<td>.69</td>
<td>.65</td>
<td>.56</td>
<td>.63</td>
<td>.62</td>
<td>.62</td>
</tr>
</tbody>
</table>

Table 1: Results (Average accuracy across all five sets of data). Bold indicates best performance for that word.

5 Conclusion and Future Work

We present an approach to unsupervised WSD that works jointly with the visual and textual domains. We showed that this multimodal approach makes gains over text-only disambiguation, and outperforms previous approaches for WSD (both text-only, and multimodal), when textual contexts are limited.

This project is still in progress, and there are many avenues for further study. We do not currently exploit collocations between lexical and visual information. Also, the bag-of-SIFT visual features that we use, while effective, have little semantic content. More structured representations over segmented image regions offer greater potential for encoding semantic content (Duygulu et al., 2002).
References


