UNSUPERVISED WORD SENSE DISAMBIGUATION RIVALING SUPERVISED METHODS (DAVID YAROWSKY, 1995)

Presentation by Victor Santos
LCT Masters – COLI (UDS)

03.5.2010 – Saarbrücken

List of contents

The paper

- Yarowsky (1995) describes an unsupervised learning algorithm for word sense disambiguation that, when trained on unannotated (untagged) English text, performs better and is simpler and less time consuming than supervised algorithms that require hand annotations.

A quick review of terminology that we will need

<u>Word-sense disambiguation</u>:

Deciding which sense of an ambiguous word is meant in a specific context.

Ex: ‘This plant has been here for only 5 months’.

Collocation (as used by Yarowsky):

A relation that holds between words that tend to appear close to each other much more frequently than randomness would predict or than observed for any random two words in a text.

Ex:

‘The **E.Ts** will come from *space* and **conquer** all of us.’

Logarithm:

The logarithm of 100 to the base 10 is 2, because

$10^2 = 100$

Supervised Learning Algorithm

- In supervised learning, we have a training data set made of data set points labeled with their respective class ($k_1 \ldots k_n$). Each point in the data set is composed of certain features ($f_1 \ldots f_n$). The goal of the algorithm is to induce/learn the correlation between the features and the classes, so that it can then apply what it learned to a new data set (test set) and correctly classify data points it has not seen before.

Example of labeled training set for money loan (class=give loan)

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

Supervised learning in the context of word-sense disambiguation in languages

- We start with a big corpus, like SemCor, for example, which is a subset of the Brown corpus and contains 234,000 words, with each open-class word in each sentence labeled with its Wordnet sense. (all labeling was done manually).
- We usually make use of two kinds of features, combining them in one of various ways: *collocational features* and *co-occurance features.*
- Collocational features (positition is important):

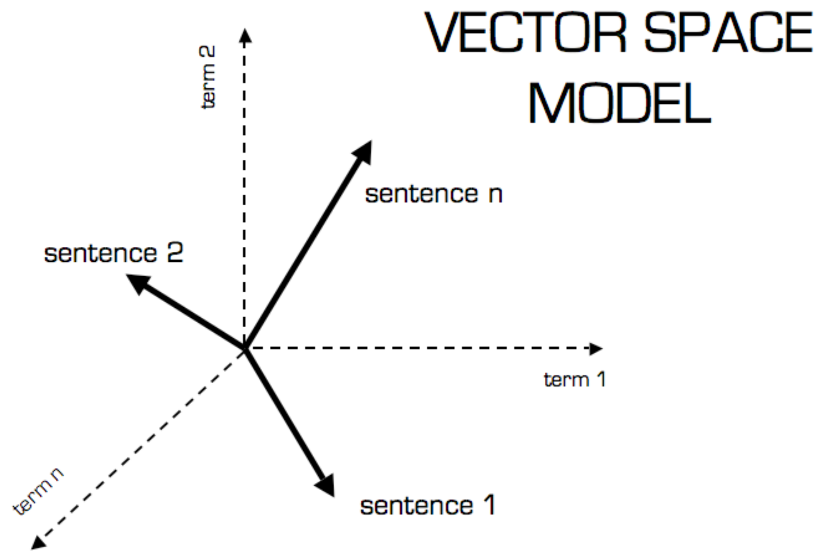
  This refers to specific words (along with their POS) which occur in a **fixed position** to the left or to the right or our target word.

Example of collocational features:

Sentence: ‘An electric guitar and **bass** player stand off to one side, ….’

A feature vector consisting of two words to the left and two words to the right of our target word (‘bass’) would result in the following vector:

[guitar, NN1, and, CJC, player, NN1, stand, VVB]



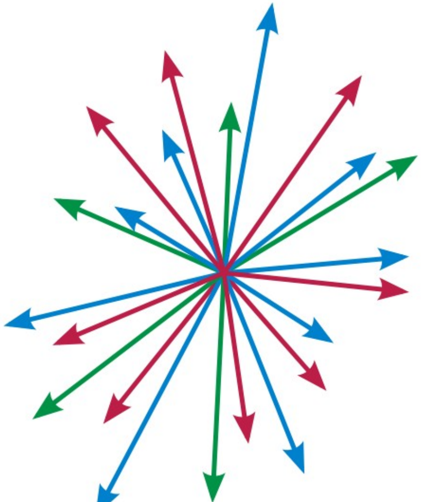
Co-occurance features:

This relates to neighboring words to our target word. In here, our features are the words themselves without their part of speech. The value of the feature is the number of times the words occur in a window surrounding the target word. For this approach to be manageable, a small number of content words frequently observed near our target word are selected as features. For the word ‘bass’, the 12 most frequent words surrounding it across many sentences in the WSJ (includes sentences from both senses of the word) are:

fishing, *big*, *sound*, *player*, *fly*, *rod*, *pound*, *double*, *runs*, *playing*, *guitar* and *band*.

Using the words above as features with a window size 10, the sentence ***‘An electric guitar and bass player stand off to one side, …’*** would be represented as the following vector:

[0,0,0,1,0,0,0,0,0,0,1,0]



Unsupervised Learning Algorithm

- In unsupervised learning, we start with a training data set which is not labeled, that is, we do not know to which class the data points belong to. All we have to start with is the features themselves and the algorithm must decide which points in the data belong to the same class. The problem is made much simpler if we know from the start the number of classes we are dealing with. We must take an initial informed guess in order to kick-start the algorithm.

Yarowsky's algorithm for word-sense disambiguation

Yarowsky's algorithm explores two powerful properties of human language, namely:

1) One sense per collocation:

Nearby words provide strong and consistent clues to the sense of a target word, conditional on relative distance, order and syntactic relationship.

Example:

'The **root** of the *plant* has **decayed**'.

'The *plant* **pesticide** has been sold for a lot of money'

'The **pesticide** *plant* has been sold for a lot of money'

2) One sense per discourse:

The sense of a target word is highly consistent within a given document. This is the first time that such a property is explored for sense-disambiguation. It's a probabilistic constraint, not a hard constraint. If the local context for another sense is strong enough, it might be overriden.

<u>Strange example:</u>

'The author J.K Rowling, a semi-vegetarian, loves eating fish. Her favorite one is the bass. Last month, she actually bought a bass, since learning to play an instrument has been a childhood dream…'

Confirmation of the OSPD hypothesis, based on 37,232 hand-tagged examples

The one-sense-per-discourse hypothesis:

Word	Senses	Accuracy	Applicblty
plant	living/factory	99.8 %	72.8 %
tank	vehicle/contnr	99.6 %	50.5 %
poach	steal/boil	100.0 %	44.4 %
palm	tree/hand	99.8 %	38.5 %
axes	grid/tools	100.0 %	35.5 %
sake	benefit/drink	100.0 %	33.7 %
bass	fish/music	100.0 %	58.8 %
space	volume/outer	99.2 %	67.7 %
motion	legal/physical	99.9 %	49.8 %
crane	bird/machine	100.0 %	49.1 %
Average		**99.8 %**	**50.1 %**

HOW THE ALGORITHM WORKS (5 STEPS)

Step 1:

In a large corpus, identify all examples of the given polysemous word, storing their contexts as lines in an initially untagged training set, as shown on above:

Sense	Training Examples (Keyword in Context)
?	... company said the *plant* is still operating
?	Although thousands of *plant* and animal species
?	... zonal distribution of *plant* life . ...
?	... to strain microscopic *plant* life from the ...
?	vinyl chloride monomer *plant* , which is ...
?	and Golgi apparatus of *plant* and animal cells
?	... computer disk drive *plant* located in ...
?	... divide life into *plant* and animal kingdom
?	... close-up studies of *plant* life and natural

Step 2

For each possible sense of the word ($k_1 \ldots k_n$), identify a small number of collocations representative of that sense and then tag all the sentences from Step 1 which contain the seed collocation with the seed's sense label. The remainder of the examples (typically 85-98%) constitute an untagged *residual*.

Sense	Training Examples (Keyword in Context)
A	used to strain microscopic ***plant*** life from the ...
A	... zonal distribution of ***plant*** **life** . ...
A	close-up studies of ***plant*** **life** and natural ...
A	too rapid growth of aquatic ***plant*** **life** in water ...
A	... the proliferation of ***plant*** and animal **life** ...
A	establishment phase of the ***plant*** virus **life** cycle ...
?	... vinyl chloride monomer ***plant*** , which is ...
?	... molecules found in ***plant*** and animal tissue
?	... Nissan car and truck ***plant*** in Japan is ...
?	... and Golgi apparatus of ***plant*** and animal cells ...
?	... union responses to ***plant*** closures . ...
?	
B	... ...
B	automated **manufacturing** ***plant*** in Fremont ...
B	... vast **manufacturing** ***plant*** and distribution ...
B	chemical **manufacturing** ***plant*** , producing viscose
B	... keep a **manufacturing** ***plant*** profitable without
B	computer **manufacturing** ***plant*** and adjacent ...
B	discovered at a St. Louis ***plant*** **manufacturing**

After Step 2

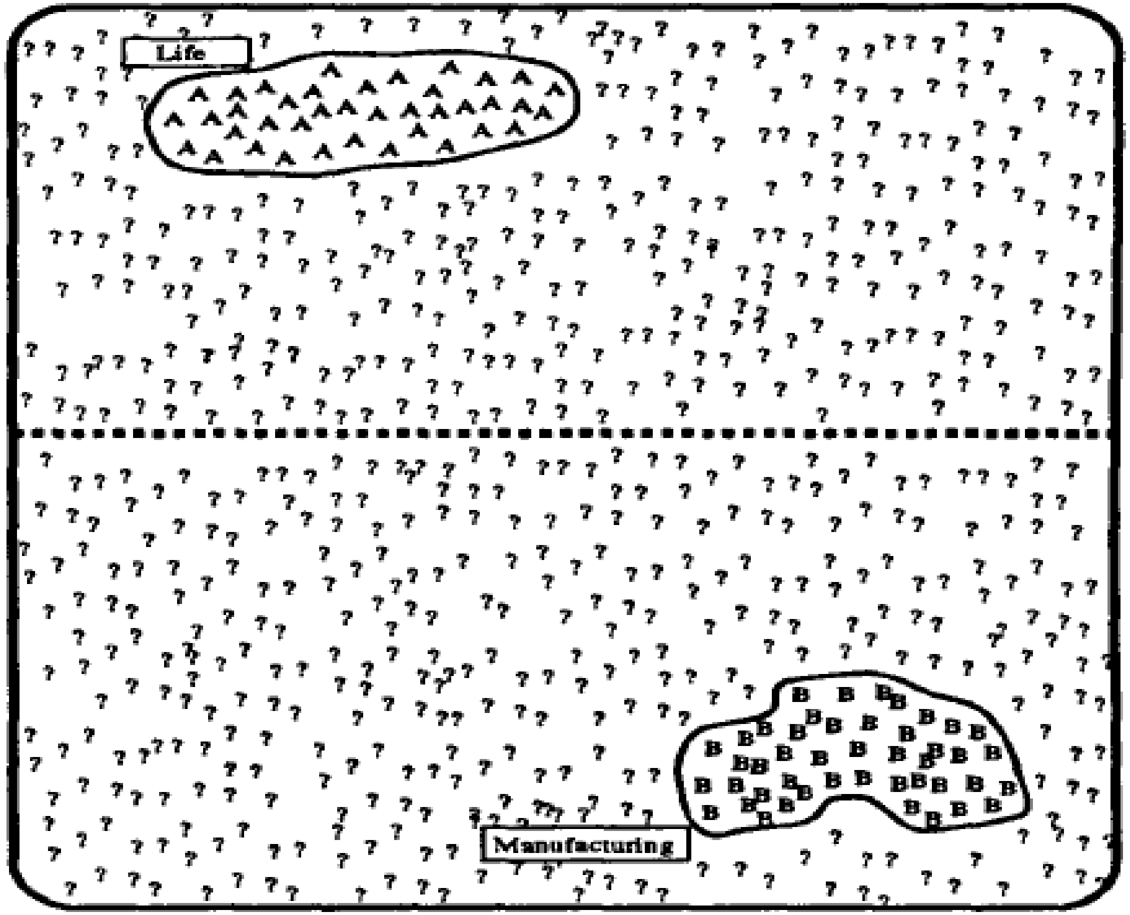


Figure 1: Sample Initial State

A = SENSE-A training example
B = SENSE-B training example
? = currently unclassified training example
Life = Set of training examples containing the collocation "life".

Step 3a (out of a-d)

Train the algorithm on the SENSE-A / SENSE-B seed sets (the residual is not used yet). The decision-list algorithm identifies other collocations that reliably partition the seed training data, ranked by the purity of the distribution.

Initial decision list for *plant* (abbreviated)

LogL	Collocation	Sense
8.10	***plant* life**	⇒ A
7.58	**manufacturing *plant***	⇒ B
7.39	life (within ±2-10 words)	⇒ A
7.20	manufacturing (in ±2-10 words)	⇒ B
6.27	animal (within ±2-10 words)	⇒ A
4.70	equipment (within ±2-10 words)	⇒ B
4.39	employee (within ±2-10 words)	⇒ B
4.30	assembly *plant*	⇒ B
4.10	*plant* closure	⇒ B
3.52	*plant* species	⇒ A
3.48	automate (within ±2-10 words)	⇒ B
3.45	microscopic *plant*	⇒ A
	...	

How LogL is calculated

$$\text{LogL} = \text{Ln}\ \frac{\Pr ob(Sense-a \mid collocation_k)}{\Pr ob(Sense-b \mid collocation_k)}$$

Step 3b (out of a-d)

Apply the resulting classifier to the entire sample set. Take those members in the residual which are tagged as SENSE-A or SENSE-B with probability above a certain threshold, and add those examples to the growing seed sets. Using the decision list, these new additions will contain new collocations reliably indicative of the previously trained seed sets.

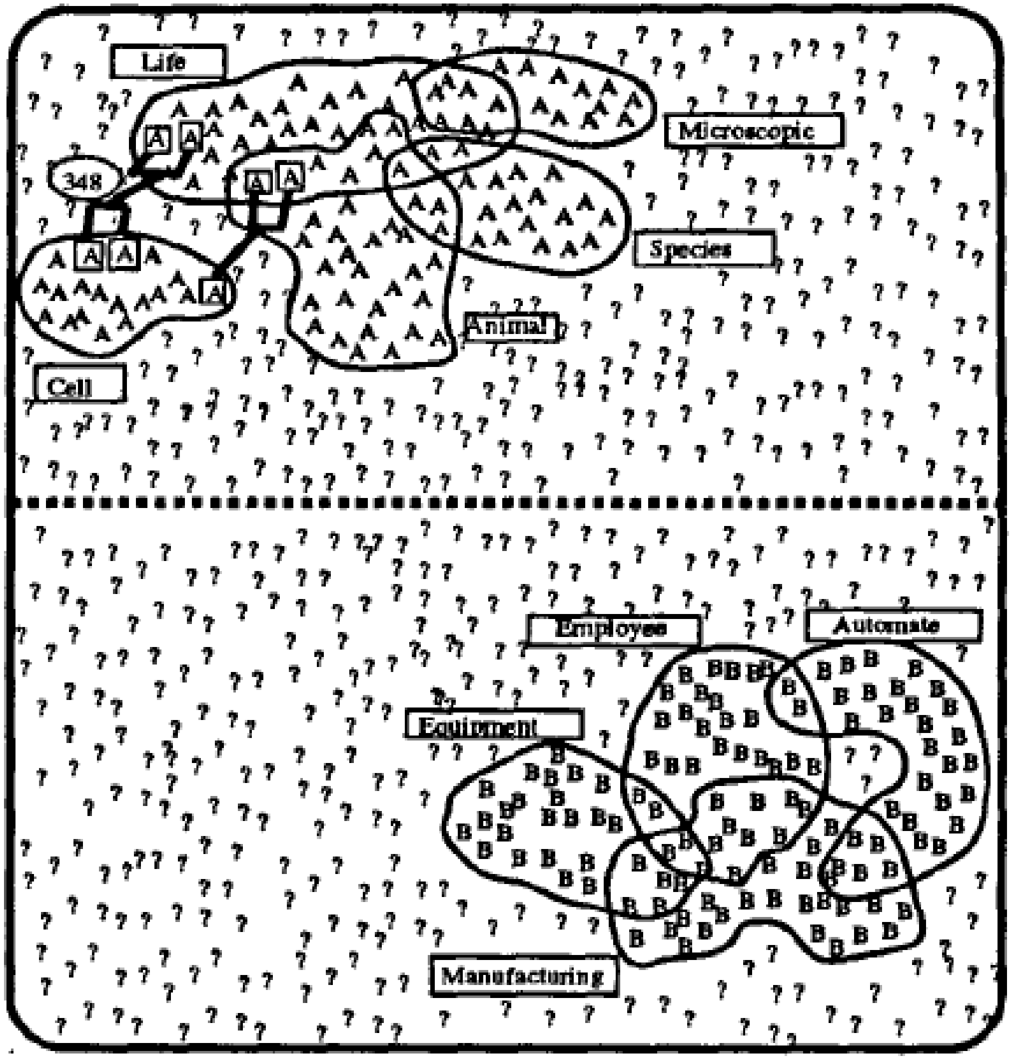


Figure 2: Sample Intermediate State
(following Steps 3b and 3c)

Important point about Step 3b

In Step 3b, when applying the decision list to previous residual sentences, there might be sentences that contain collocations from both classes at the same time (Sense A and Sense B), for example:

> ‘An **employee** (Sense-B, LogL4.39) whose **animal** (Sense-A, Log 6.27) ate a dangerous *plant* damaged the **equipment** (Sense-B, LogL4.70)’.

Only the most predictable collocation is taken into account for deciding the sense of the polysemous word. In this case, it will be tag as **SENSE-A.**

Step 3c (out of a-d) The one-sense-per-discourse step

This is the step where the *one-sense-per discourse* tendency comes into play. It is used to both augment (increase) the training set or to correct (filter) erroneously labeled examples. It is important to point out this is conditional on the relative numbers and the probabilities associated with the tagged examples in the discourse.

Examples (next slide):

The augmentation use of Step 3c

Labeling previously untagged contexts
using the one-sense-per-discourse property

Change in tag	Disc. Numb.	Training Examples (from same discourse)
A → A	724	... the existence of *plant* and animal life ...
A → A	724	... classified as either *plant* or animal ...
? → A	724	Although bacterial and *plant* cells are enclosed

In this example, we can see that the third sentence in the discourse has no collocation previously identified before. However, given the one-sense-per-discourse ‘rule’, we can label it and therefore augment our training set. This works as a bridge to new collocations (in this case, the collocation ‘cell/cells’.

The filter (error correction) use of Step 3c

Error Correction using the one-sense-per-discourse property

Change in tag	Disc. Numb.	Training Examples (from same discourse)
A → A	525	contains a varied *plant* and animal life
A → A	525	the most common *plant* life , the ...
A → A	525	slight within Arctic *plant* species ...
B → A	525	are protected by *plant* parts remaining from

We can see here that even though the fourth sentence in this discourse had been labeled as sense B, due to the one-sense-per-discourse law, we decide that it should actually belong to SENSE-A, instead of SENSE-B.

Step 3d (out of a-d)
The iterative step

Repeat Step 3a-3c iteratively. The training set, i.e., those sentences with occurances of the polysemous word Labeled either SENSE-A or SENSE-B will tend to grow, while the residual (occurrences of the word which have not yet been labeled) will tend to shrink.

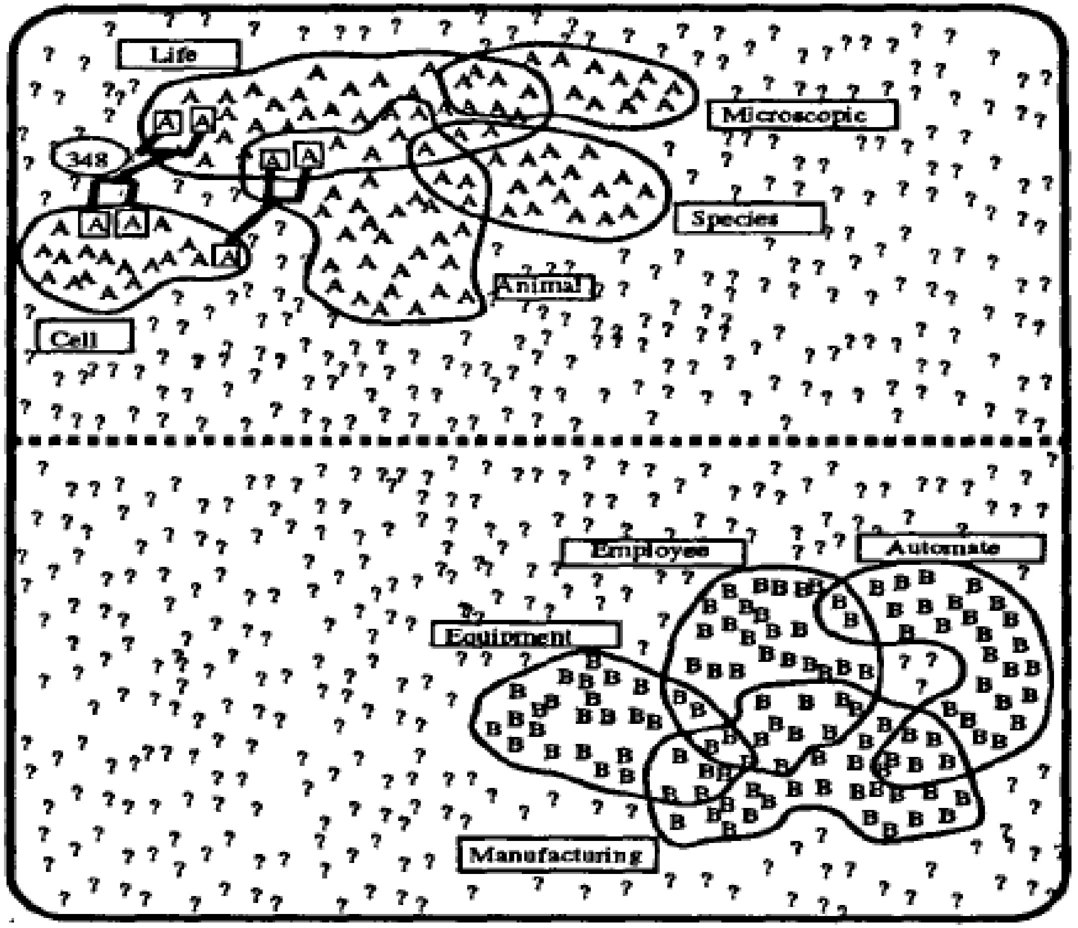


Figure 2: Sample Intermediate State
(following Steps 3b and 3c)

STEP 4

Stop. When the training parameters are held constant, the algorithm will converge on a stable residual set. Reminder: Even though most training examples will exhibit multiple collocations indicative of the same sense, only the highest Log actually influences our choice for what sense to assign (this circumvents problems associated with non-independent evidence sources).

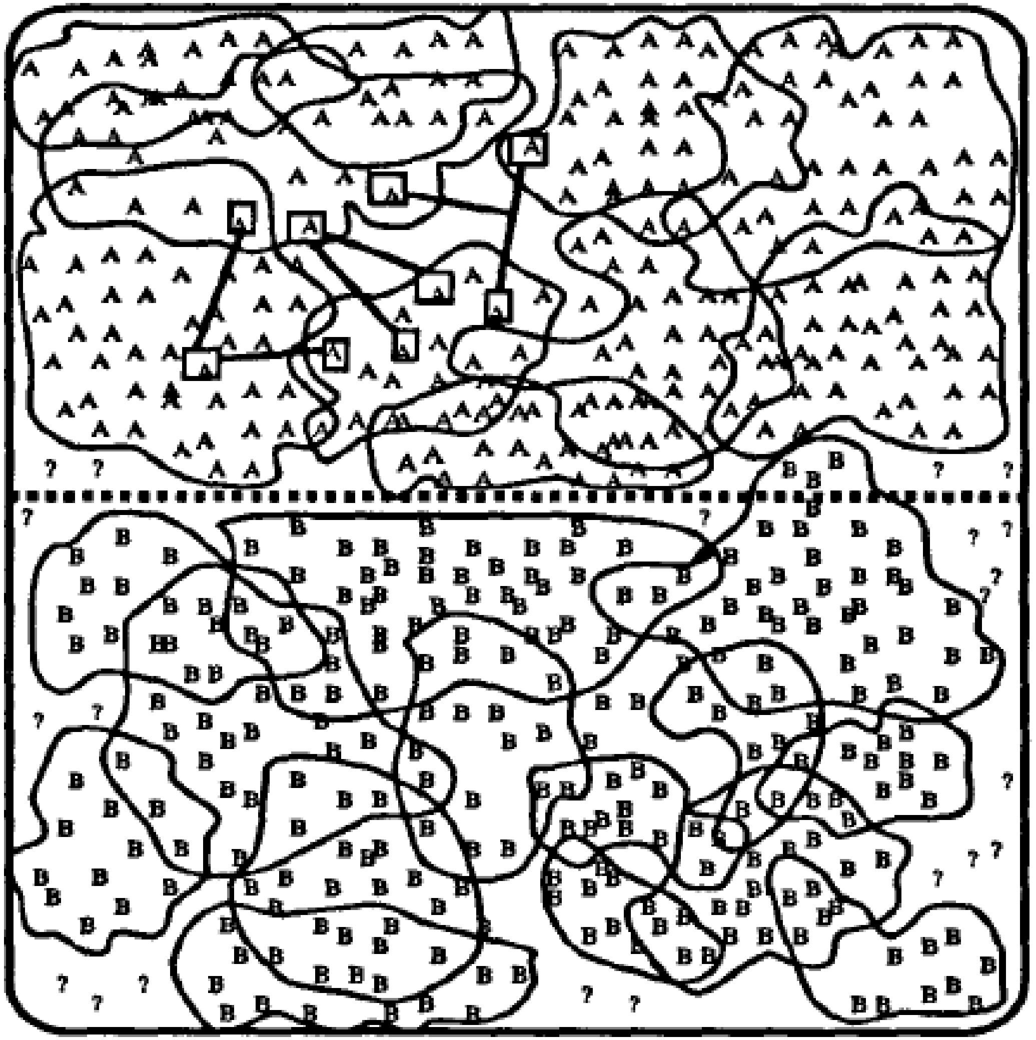


Figure 3: Sample Final State

STEP 5

After completing steps 1-4, we can now apply the classifier to new data and/or use it to annotate the original untagged corpus with sense tags and probabilities.

Notice that the initial decision list is quite different from the final one.

Initial decision list for *plant* (abbreviated)

LogL	Collocation	Sense
8.10	*plant* life	⇒ A
7.58	manufacturing *plant*	⇒ B
7.39	life (within ±2-10 words)	⇒ A
7.20	manufacturing (in ±2-10 words)	⇒ B
6.27	animal (within ±2-10 words)	⇒ A
4.70	equipment (within ±2-10 words)	⇒ B
4.39	employee (within ±2-10 words)	⇒ B
4.30	assembly *plant*	⇒ B
4.10	*plant* closure	⇒ B
3.52	*plant* species	⇒ A
3.48	automate (within ±2-10 words)	⇒ B
3.45	microscopic *plant*	⇒ A
	...	

Final decision list for *plant* (abbreviated)

LogL	Collocation	Sense
10.12	*plant* growth	⇒ A
9.68	car (within $\pm k$ words)	⇒ B
9.64	*plant* height	⇒ A
9.61	union (within $\pm k$ words)	⇒ B
9.54	equipment (within $\pm k$ words)	⇒ B
9.51	assembly *plant*	⇒ B
9.50	nuclear *plant*	⇒ B
9.31	flower (within $\pm k$ words)	⇒ A
9.24	job (within $\pm k$ words)	⇒ B
9.03	fruit (within $\pm k$ words)	⇒ A
9.02	*plant* species	⇒ A
...	...	

Evaluation

- Data extracted from a 460 million word corpus containing news articles, scientific abstracts, spoken transcripts and novels, constituting almost certainly the largest training/testing sets used in the sense-disambiguation literature.
- Performance of multiple models compared with:

  - supervised decision lists

  - unsupervised learning algorithm by

  Schütze(1992), based on alignment of clusters with words senses and taking the bag-of-words point of view.

EVALUATION

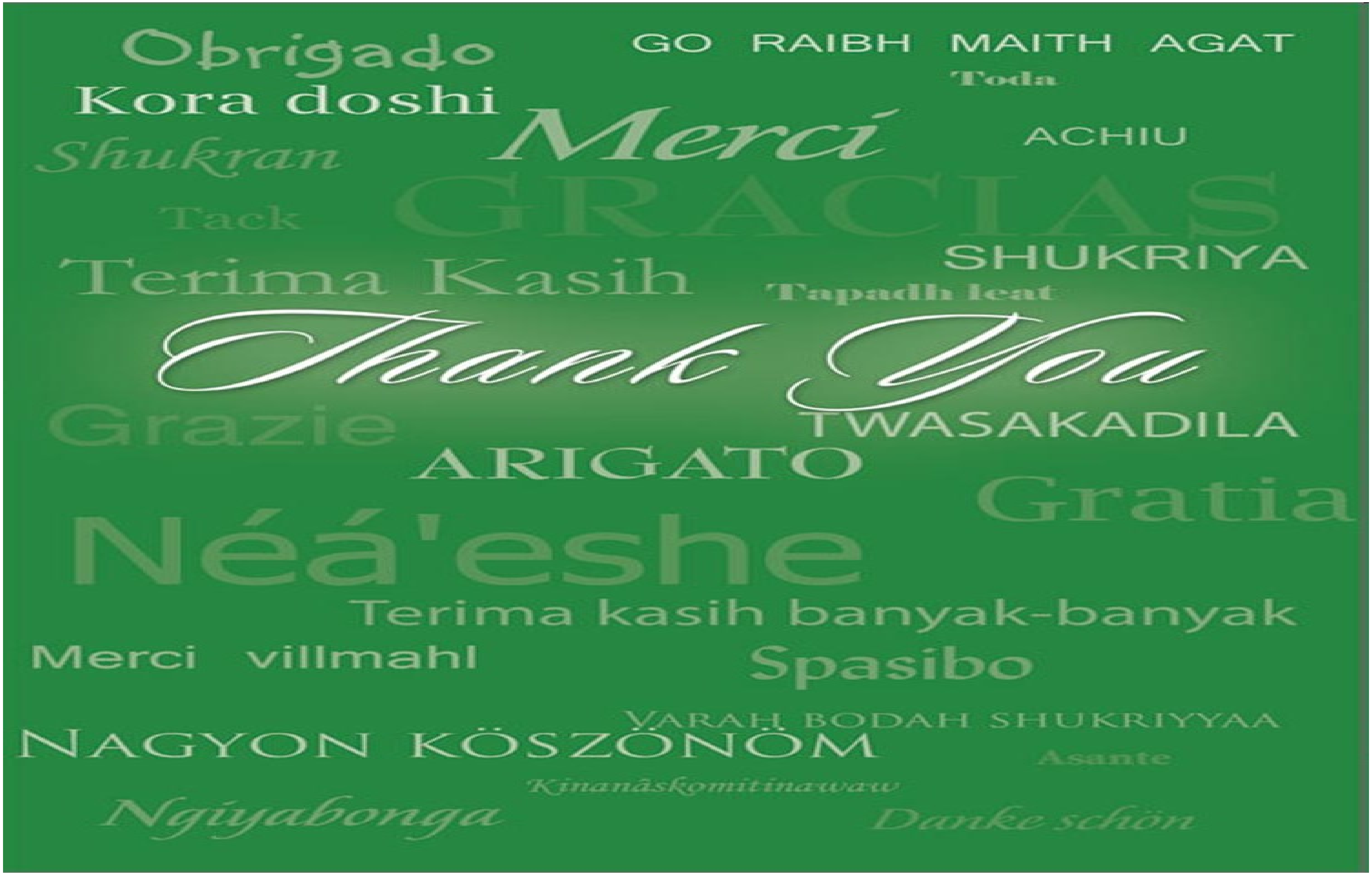
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
			%		Seed Training Options			(7) + OSPD		
Word	Senses	Samp. Size	Major Sense	Supvsd Algrtm	Two Words	Dict. Defn.	Top Colls.	End only	Each Iter.	Schütze Algrthm
plant	living/factory	7538	53.1	97.7	97.1	97.3	97.6	98.3	98.6	92
space	volume/outer	5745	50.7	93.9	89.1	92.3	93.5	93.3	93.6	90
tank	vehicle/container	11420	58.2	97.1	94.2	94.6	95.8	96.1	96.5	95
motion	legal/physical	11968	57.5	98.0	93.5	97.4	97.4	97.8	97.9	92
bass	fish/music	1859	56.1	97.8	96.6	97.2	97.7	98.5	98.8	–
palm	tree/hand	1572	74.9	96.5	93.9	94.7	95.8	95.5	95.9	–
poach	steal/boil	585	84.6	97.1	96.6	97.2	97.7	98.4	98.5	–
axes	grid/tools	1344	71.8	95.5	94.0	94.3	94.7	96.8	97.0	–
duty	tax/obligation	1280	50.0	93.7	90.4	92.1	93.2	93.9	94.1	–
drug	medicine/narcotic	1380	50.0	93.0	90.4	91.4	92.6	93.3	93.9	–
sake	benefit/drink	407	82.8	96.3	59.6	95.8	96.1	96.1	97.5	–
crane	bird/machine	2145	78.0	96.6	92.3	93.6	94.2	95.4	95.5	–
AVG		3936	63.9	96.1	90.6	94.8	95.5	96.1	96.5	92.2

*Column 11 shows Schütze's unsupervised algorithm (bag-of-words) applied to some of these words, trained on the New York Times News Service corpus. His algorithm works with clustering based on distributional parameters and he might have 10 different clusters for only 2 senses, which have to be hand-inspected at the end to decide on the sense)

*Column 5 shows the results for supervised training using the decision list algorithm, applied to the same data and not using any discourse information (OSPD).

CONCLUSION

- The algorithm works by harnessing several powerful, empirically-observed properties of language, namely the strong tendency for words to exhibit only on sense per collocation and per discourse.
- It attempts to derive maximal leverage from these properties by modeling a rich diversity of collocational relationships. It thus uses more discriminating information than available to algorithms treating documents as bag of words.
- For an unsupervised algorithm it works surprisingly well, directly outperforming Schütze’s unsupervised algorithm 96.7% to 92.2%, on a test of the same 4 words. More impressively, it achieves nearly the same performance as the supervised algorithm given identical training contexts (95.5% vs. 96.1%), and in some cases actually achieves superior performance when using the one-sense-per discourse contraint (96.5% vs. 96.1%).


Obrigado
GO RAIBH MAITH AGAT
Toda
Kora doshi
Merci
ACHIU
Shukran
GRACIAS
Tack
SHUKRIYA
Terima Kasih
Tapadh leat
Thank You
Grazie
TWASAKADILA
ARIGATO
Gratia
Néá'eshe
Terima kasih banyak-banyak
Merci villmahl
Spasibo
VARAH BODAH SHUKRIYYAA
NAGYON KÖSZÖNÖM
Asante
Kinanâskomitinawaw
Ngiyabonga
Danke schön