SenseSpotting: Never let your parallel data tie you to an old domain

Marine Carpuat[1], **Hal Daumé III**[2], **Katharine Henry**[3],
Ann Irvine[4], **Jagadeesh Jagarlamudi**[5], **Rachel Rudinger**[6]
[1] National Research Council Canada, marine.carpuat@nrc.gc.ca
[2] CLIP, University of Maryland, me@hal3.name
[3] CS, University of Chicago, kehenry@uchicago.edu
[4] CLSP, Johns Hopkins University, anni@jhu.edu
[5] IBM T.J. Watson Research Center, jags@us.ibm.com
[6] CLSP, Johns Hopkins University, rachel.rudinger@aya.yale.edu

Abstract

Words often gain new senses in new domains. Being able to automatically identify, from a corpus of monolingual text, which word *tokens* are being used in a previously unseen sense has applications to machine translation and other tasks sensitive to lexical semantics. We define a task, SenseSpotting, in which we build systems to spot tokens that have new senses in new domain text. Instead of difficult and expensive annotation, we build a gold-standard by leveraging cheaply available parallel corpora, targeting our approach to the problem of domain adaptation for machine translation. Our system is able to achieve F-measures of as much as 80%, when applied to word types it has *never seen before*. Our approach is based on a large set of novel features that capture varied aspects of how words change when used in new domains.

1 Introduction

As Magnini et al. (2002) observed, the domain of the text that a word occurs in is a useful signal for performing word sense disambiguation (e.g. in a text about finance, *bank* is likely to refer to a financial institution while in a text about geography, it is likely to refer to a *river bank*). However, in the classic WSD task, ambiguous word types and a set of possible senses are known in advance. In this work, we focus on the setting where we observe texts in two different domains and want to identify words in the second text that have a sense that did not appear in the first text, without any lexical knowledge in the new domain.

To illustrate the task, consider the French noun *rapport*. In the parliament domain, this means (and is translated as) "report." However, in moving to a medical or scientific domain, the word *gains a new sense*: "ratio", which simply does not exist in the parliament domain. In a science domain, the "report" sense exists, but it is dominated about 12:1 by "ratio." In a medical domain, the "report" sense remains dominant (about 2:1), but the new "ratio" sense appears frequently.

	état	*rapport*	*régime*
Govt.	geo. state	report	(political) regime
Medical	state (mind) geo. state	report ratio	diet (political) regime
Science	geo. state	ratio report	(political) regime diet
Movies	geo. state	report	(political) regime diet

Table 1: Examples of French words and their most frequent senses (translations) in four domains.

In this paper we define a new task that we call SenseSpotting. The goal of this task is to identify words in a new domain monolingual text that appeared in old domain text but which have a new, previously unseen sense[1]. We operate under the framework of phrase sense disambiguation (Carpuat and Wu, 2007), in which we take automatically align *parallel* data in an old domain to generate an initial old-domain sense inventory. This sense inventory provides the set of "known" word senses in the form of phrasal translations. Concrete examples are shown in Table 1. One of our key contributions is the development of a rich set of features based on monolingual text that are indicative of new word senses.

This work is driven by an application need. When machine translation (MT) systems are applied in a new domain, many errors are a result of: (1) previously unseen (OOV) source language words, or (2) source language words that appear with a new sense and which require new transla-

[1] All features, code, data and raw results are at: github.com/hal3/IntrinsicPSDEvaluation

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1435–1445,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

tions[2] (Carpuat et al., 2012). Given monolingual text in a new domain, OOVs are easy to identify, and their translations can be acquired using dictionary extraction techniques (Rapp, 1995; Fung and Yee, 1998; Schafer and Yarowsky, 2002; Schafer, 2006; Haghighi et al., 2008; Mausam et al., 2010; Daumé III and Jagarlamudi, 2011), or active learning (Bloodgood and Callison-Burch, 2010). However, previously seen (even frequent) words which require new translations are harder to spot.

Because our motivation is translation, one significant point of departure between our work and prior related work (§3) is that we focus on *word tokens*. That is, we are not interested *only* in the question of "has this known word (type) gained a new sense?", but the much more specific question of "is this particular (token) *occurrence* of this known word being used in a new sense?" Note that for both the dictionary mining setting *and* the active learning setting, it is important to consider words in context when acquiring their translations.

2 Task Definition

Our task is defined by two data components. Details about their creation are in §5. First, we need an old-domain sense dictionary, extracted from French-English parallel text (in our case, parliamentary proceedings). Next, we need new-domain monolingual French text (we use medical text, scientific text and movie subtitle text). Given these two inputs, our challenge is to find tokens in the new-domain text that are being used in a new sense (w.r.t. the old-domain dictionary).

We assume that we have access to a small amount of new domain parallel "tuning data." From this data, we can extract a small new domain dictionary (§5). By comparing this new domain dictionary to the old domain dictionary, we can identify which words have gained new senses. In this way, we turn the SENSESPOTTING problem into a supervised binary classification problem: an example is a French word in context (in the new domain monolingual text) and its label is positive when it is being used in a sense that did not exist in the old domain dictionary. In this task, the classifier is always making predictions on words *outside* this tuning data on word types it *has never seen before!* From an applied perspective, the assumption of a small amount of parallel data in the new domain is reasonable: if we want an MT system for a new domain, we will likely have some data for system tuning and evaluation.

3 Related Work

While word senses have been studied extensively in lexical semantics, research has focused on word sense disambiguation, the task of disambiguating words in context given a predefined sense inventory (e.g., Agirre and Edmonds (2006)), and word sense induction, the task of learning sense inventories from text (e.g., Agirre and Soroa (2007)). In contrast, detecting novel senses has not received as much attention, and is typically addressed within word sense induction, rather than as a distinct SENSESPOTTING task. Novel sense detection has been mostly motivated by the study of language change over time. Most approaches model changes in co-occurrence patterns for *word types* when moving between corpora of old and modern language (Sagi et al., 2009; Cook and Stevenson, 2010; Gulordava and Baroni, 2011).

Since these type-based models do not capture polysemy in the new language, there have been a few attempts at detecting new senses at the token-level as in SENSESPOTTING. Lau et al. (2012) leverage a common framework to address sense induction and disambiguation based on topic models (Blei et al., 2003). Sense induction is framed as learning topic distributions for a word type, while disambiguation consists of assigning topics to word tokens. This model can interestingly be used to detect newly coined senses, which might co-exist with old senses in recent language. Bamman and Crane (2011) use parallel Latin-English data to learn to disambiguate Latin words into English senses. New English translations are used as evidence that Latin words have shifted sense. In contrast, the SENSESPOTTING task consists of detecting when senses are unknown in parallel data.

Such novel sense induction methods require manually annotated datasets for the purpose of evaluation. This is an expensive process and therefore evaluation is typically conducted on a very small scale. In contrast, our SENSESPOTTING task leverages automatically word-aligned parallel corpora as a source of annotation for supervision during training and evaluation.

[2] Sense shifts do not always demand new translations; some ambiguities are preserved across languages. E.g., *fenêtre* can refer to a window of a building or on a monitor, but translates as "window" either way. Our experiments use bilingual data with an eye towards improving MT performance: we focus on words that demand new translations.

The impact of domain on novel senses has also received some attention. Most approaches operate at the *type*-level, thus capturing changes in the most frequent sense of a word when shifting domains (McCarthy et al., 2004; McCarthy et al., 2007; Erk, 2006; Chan and Ng, 2007). Chan and Ng (2007) notably show that detecting changes in predominant sense as modeled by domain sense priors can improve sense disambiguation, even after performing adaptation using active learning.

Finally, SENSESPOTTING has not been addressed directly in MT. There has been much interest in translation mining from parallel or comparable corpora for *unknown* words, where it is easy to identify which words need translations. In contrast, SENSESPOTTING detects when words have new senses and, thus, frequently a new translation. Work on active learning for machine translation has focused on collecting translations for longer unknown segments (e.g., Bloodgood and Callison-Burch (2010)). There has been some interest in detecting which phrases that are hard to translate for a given system (Mohit and Hwa, 2007), but difficulties can arise for many reasons: SENSESPOTTING focuses on a single problem.

4 New Sense Indicators

We define features over both *word types* and *word tokens*. In our classification setting, each instance consists of a French word token in context. Our *word type* features ignore this context and rely on statistics computed over our entire new domain corpus. In contrast, our *word token* features consider the context of the particular instance of the word. If it were the case that only one sense existed for all word tokens of a particular type within a single domain, we would expect our word type features to be able to spot new senses without the help of the word token features. However, in fact, even within a single domain, we find that often a word type is used with several senses, suggesting that word token features may also be useful.

4.1 Type-level Features

Lexical Item Frequency Features A very basic property of the new domain that we hope to capture is that word frequencies change, and such changes might be indicative of a domain shift. As such, we compute unigram log probabilities (via smoothed relative frequencies) of each word under consideration in the old domain and the new domain. We then add as features these two log probabilities as well as their difference. These are our Type:RelFreq features.

N-gram Probability Features The goal of the Type:NgramProb feature is to capture the fact that "unusual contexts" might imply new senses. To capture this, we can look at the log probability of the word under consideration given its N-gram context, both according to an old-domain language model (call this $\ell^{\text{old}}_{\text{ng}}$) and a new-domain language model (call this $\ell^{\text{new}}_{\text{ng}}$). However, we do not simply want to capture unusual words, but words that are unlikely in context, so we also need to look at the respective unigram log probabilities: $\ell^{\text{old}}_{\text{ug}}$ and $\ell^{\text{new}}_{\text{ug}}$. From these four values, we compute corpus-level (and therefore type-based) statistics of the new domain n-gram log probability ($\ell^{\text{new}}_{\text{ng}}$, the difference between the n-gram probabilities in each domain ($\ell^{\text{new}}_{\text{ng}} - \ell^{\text{old}}_{\text{ng}}$), the difference between the n-gram and unigram probabilities in the new domain ($\ell^{\text{new}}_{\text{ng}} - \ell^{\text{new}}_{\text{ug}}$), and finally the combined difference: $\ell^{\text{new}}_{\text{ng}} - \ell^{\text{new}}_{\text{ug}} + \ell^{\text{old}}_{\text{ug}} - \ell^{\text{old}}_{\text{ng}}$). For each of these four values, we compute the following type-based statistics over the monolingual text: mean, standard deviation, minimum value, maximum value and sum. We use trigram models.

Topic Model Feature The intuition behind the topic model feature is that if a word's distribution over topics changes when moving into a new domain, it is likely to also gain a new sense. For example, suppose that in our old domain, the French word *enceinte* is only used with the sense "wall," but in our new domain, *enceinte* may have senses corresponding to either "wall" or to "pregnant." We would expect to see this reflected in *enceinte*'s distribution over topics: the topic that places relatively high probabilities on words such as "bébé" (English "baby") and *enfant* (English "child") will also place a high probability on *enceinte* when trained on new domain data. In the old domain, however, we would not expect a similar topic (if it exists) to give a high probability to *enceinte*. Based on this intuition, for all words w, where T_o and T_n are the set of old and new topics and P_o and P_n are the old and new distributions defined over them, respectively, and *cos* is the cosine similarity between a pair of topics, we define the feature Type:TopicSim: $\sum_{t \in T_n, t' \in T_o} P_n(t|w) P_o(t'|w) \cos(t, t')$. For a word w, the feature value will be high if, for each new domain topic t that places high probability on w, there is an old domain topic t' that is similar to t and also places a high probability on w. Conversely, if no such topic exists, the score will be low, indicating the word has gained a new sense. We use the online LDA (Blei et al., 2003; Hoffman et al., 2010), implemented in `http://hunch.net/~vw/` to compute topics on the two domains separately. We use 100 topics.

Context Feature It is expected that words acquiring new senses will tend to neighbor different sets of words (e.g. different arguments, prepositions, parts of speech, etc.). Thus, we define an additional type level feature to be the ratio of the number of new domain n-grams (up to length three) that contain word w and which do not appear in the old domain to the total number of new domain n-grams containing w. With N_w indicating the set of n-grams in the new domain which contain w, O_w indicating the set of n-grams in the old domain which contain w, and $|N_w - O_w|$ indicating the n-grams which contain w and appear in the new but not the old domain, we define Type:Contextas $\frac{|N_w - O_w|}{|N_w|}$. We do not count n-grams containing OOVs, as they may simply be instances of applying the same sense of a word to a new argument

4.2 Token-level Features

N-gram Probability Features Akin to the N-gram probability features at the type level (namely, Token:NgramProb), we compute the same values at the token level (new/old domain and unigram/trigram). Instead of computing statistics over the entire monolingual corpus, we use the instantaneous values of these features for the token under consideration. The six features we construct are: unigram (and trigram) log probabilities in the old domain, the new domain, and their difference.

Context Features Following the type-level n-gram feature, we define features for a particular word *token* based on its n-gram context. For token w_i, in position i in a given sentence, we consider its context words in a five word window: w_{i-2}, w_{i-1}, w_{i+1}, and w_{i+2}. For each of the four contextual words in positions $p = \{-2, -1, 1, 2\}$, relative to i, we define the following feature, Token:CtxCnt: $\log(c_{w_p})$ where c_{w_p} is the number of times word w_p appeared in position p relative to w_i in the OLD-domain data. We also define a single feature which is the percent of the four contextual words which had been seen in the OLD-domain data, Token:Ctx%.

Token-Level PSD Features These features aim to capture generalized characteristics of a context. Towards this end, first, we pose the problem as a phrase sense disambiguation (PSD) problem over the known sense inventory. Given a source word in a context, we train a classifier to predict the most likely target translation. The ground truth labels (target translation for a given source word) for this classifier are generated from the phrase table of the old domain data. We use the same set of features as in Carpuat and Wu (2007). Second, given a source word s, we use this classifier to compute the probability distribution of target translations $\big(p(t|s)\big)$. Subsequently, we use this probability distribution to define new features for the SENSESPOTTING task. The idea is that, if a word is used in one of the known senses then its context must have been seen previously and hence we hope that the PSD classifier outputs a spiky distribution. On the other hand, if the word takes a new sense then hopefully it is used in an unseen context resulting in the PSD classifier outputting an uniform distribution. Based on this intuition, we add the following features: **MaxProb** is the maximum probability of any target translation: $\max_t p(t|s)$. **Entropy** is the entropy of the probability distribution: $-\sum_t p(t|s) \log p(t|s)$. **Spread** is the difference between maximum and minimum probabilities of the probability distribution: $\big(\max_t p(t|s) - \min_t p(t|s)\big)$. **Confusion** is the uncertainty in the most likely prediction given the source token: $\frac{\text{median}_t p(t|s)}{\max_t p(t|s)}$. The use of median in the numerator rather than the second best is motivated by the observation that, in most cases, top ranked translations are of the same sense but differ in morphology.

We train the PSD classifier in two modes: 1) a single global classifier that predicts the target translation given any source word; 2) a local classifier for each source word. When training the global PSD classifier, we include some lexical features that depend on the source word. For both modes, we use real valued and binned features giving rise to four families of features Token:G-PSD, Token:G-PSDBin, Token:L-PSD and Token:L-PSDBin.

Prior vs. Posterior PSD Features When the PSD classifier is trained in the second mode, i.e. one classifier per word type, we can define additional features based on the prior (with out the word context) and posterior (given the word's context) probability distributions output by the classifier, i.e. $p_{\text{prior}}(t|s)$ and $p_{\text{post.}}(t|s)$ respec-

Domain	Sentences	Lang	Tokens	Types
Hansard	8,107,356	fr	161,695,309	191,501
		en	144,490,268	186,827
EMEA	472,231	fr	6,544,093	34,624
		en	5,904,296	29,663
Science	139,215	fr	4,292,620	117,669
		en	3,602,799	114,217
Subs	19,239,980	fr	154,952,432	361,584
		en	174,430,406	293,249

Table 2: Basic characteristics of the parallel data.

	Parallel		Repr.	Repr.	% New
	Sents	fr-tok	Types	Tokens	Sense
EMEA	24k	270k	399	35,266	52.0%
Science	22k	681k	425	8,355	24.3%
Subs	36k	247k	388	22,598	43.4%

Table 3: Statistics about representative words and the size of the development sets. The columns show: the total amount of parallel development data (# of sentences and tokens in French), # of representative types that appear in this corpus, the corresponding # of tokens, and the percentage of these tokens that correspond to "new senses."

tively. We compute the following set of features referred to as Token:PSDRatio: **SameMax** checks if both the prior and posterior distributions have the same translation as the most likely translation. **SameMin** is same as the above feature but check if the least likely translation is same. **X-OR_MinMax** is the exclusive-OR of **SameMax** and **SameMin** features. **KL** is the KL-divergence between the two distributions. Since KL-divergence is asymmetric, we use $KL(p_{\text{prior}}||p_{\text{post.}})$ and $KL(p_{\text{post.}}||p_{\text{prior}})$. **MaxNorm** is the ratio of maximum probabilities in prior and posterior distributions. **SpreadNorm** is the ratio of spread of the prior and posterior distributions, where spared is the difference between maximum and minimum probabilities of the distribution as defined earlier. **ConfusionNorm** is the ratio of confusion of the prior and posterior distributions, where confusion is defined as earlier.

5 Data and Gold Standard

The first component of our task is a parallel corpus of old domain data, for which we use the French-English Hansard parliamentary proceedings (`http://www.parl.gc.ca`). From this, we extract an old domain sense dictionary, using the Moses MT framework (Koehn et al., 2007). This defines our old domain sense dictionary. For new domains, we use three sources: (1) the EMEA medical corpus (Tiedemann, 2009), (2) a corpus of scientific abstracts, and (3) a corpus of translated movie subtitles (Tiedemann, 2009). Basic statistics are shown in Table 2. In all parallel corpora, we normalize the English for American spelling.

To create the gold standard *truth*, we followed a lexical sample apparoach and collected a set of 300 "representative types" that are interesting to evaluate on, because they have multiple senses within a single domain or whose senses are likely to change in a new domain. We used a semi-automatic approach to identify representative types. We first used the phrase table from the Moses output to rank phrases in each domain using TF-IDF scores with Okapi BM25 weighting. For each of the three new domains (EMEA, Science, and Subs), we found the intersection of phrases between the old and the new domain. We then looked at the different translations that each had in the phrase table and a French speaker selected a subset that have multiple senses.[3]

In practice, we limited our set almost entirely to source *words*, and included only a single multi-word phrase, *vue des enfants*, which usually translates as "for children" in the old domain but almost always translates as "sight of children" in the EMEA domain (as in "...should be kept out of *the sight of children*"). Nothing in the way we have defined, approached, or evaluated the SENSESPOTTING task is dependent on the use of representative words instead of longer representative phrases. We chose to consider mostly source language words for simplicity and because it was easier to identify good candidate words.

In addition to the manually chosen words, we also identified words where the translation with the highest lexical weight varied in different domains, with the intuition being that are the words that are likely to have acquired a new sense. The top 200 words from this were added to the manually selected representative words to form a list of 450. Table 3 shows some statistics about these words across our three test domains.

6 Experiments

6.1 Experimental setup

Our goal in evaluation is to be able to understand what our approach is realistically capable of. One challenge is that the distribution

[3] In order to create the *evaluation data*, we used both sides of the full parallel text; we do *not* use the English side of the parallel data for actually building systems.

of representative words is highly skewed.[4] We present results in terms of area under the ROC curve (AUC),[5] micro-averaged precision/recall/f-measure and macro-averaged precision/recall/f-measure. For macro-averaging, we compute a single confusion matrix over all the test data and determining P/R/F from that matrix. For micro-averaging, we compute a separate confusion matrix *for each word type* on the French side, compute P/R/F for each of these separately, and then average the results. (Thus, micro-F is not a function of micro-P and micro-R.) The AUC and macro-averaged scores give a sense of how well the system is doing on a type-level basis (essentially weighted by type frequency), while the micro-averaged scores give a sense as to how well the system is doing on individual types, not taking into account their frequencies.

For most of our results, we present standard deviations to help assess significance ($\pm 2\sigma$ is roughly a 90% confidence interval). For our results, in which we use new-domain training data, we compute these results via 16-fold cross validation. The folds are split *across types* so *the system is never being tested on a word type that it has seen before*. We do this because it more closely resembles our application goals. We do 16-fold for convenience, because we divide the data into binary folds recursively (thus having a power-of-two is easier), with an attempt to roughly balance the size of the training sets in each fold (this is tricky because of the skewed nature of the data). This entire 16-fold cross-validation procedure is repeated 10 times and averages and standard deviations are over the 160 replicates.

We evaluate performance using our type-level features only, TYPEONLY, our token-level features only, TOKENONLY, and using both our type and our token level features, ALLFEATURES.

We compare our results with two baselines: RANDOM and CONSTANT. RANDOM predicts new-sense or not-new-sense randomly and with equal probability. CONSTANT always predicts new-sense, achieving 100% recall and a macro-level precision that is equal to the percent of representative words which do have a new sense, modulo cross-validation splits (see Table 3). Additionally, we compare our results with a type-level oracle, TYPEORACLE. For all tokens of a given word type, the oracle predicts the majority label (new-sense or not-new-sense) for that word type. These results correspond to an upper bound for the TYPEONLY experiments.

6.2 Classification Setup

For all experiments, we use a linear classifier trained by stochastic gradient descent to optimize logistic loss. We also did some initial experiments on development data using boosted decision trees instead and other loss functions (hinge loss, squared loss), but they never performed as well. In all cases, we perform 20 passes over the training data, using development data to perform early stopping (considered at the end of each pass). We also use development data to tune a regularizer (either ℓ_1 or ℓ_2) and its regularization weight.[6] Finally, all real valued features are automatically bucketed into 10 consecutive buckets, each with (approximately) the same number of elements. Each learner uses a small amount of development data to tune a threshold on scores for predicting new-sense or not-a-new-sense, using macro F-measure as an objective.

6.3 Result Summary

Table 4 shows our results on the SENSESPOTTING task. Classifiers based on the features that we defined outperform both baselines in all macro-level evaluations for the SENSESPOTTING task. Using AUC as an evaluation metric, the TOKENONLY, TYPEONLY, and ALLFEATURES models performed best on EMEA, Science, and Subtitles data, respectively. Our token-level features perform particularly poorly on the Science and Subtitles data. Although the model trained on only those features achieves reasonable precision (72.59 and 70.00 on Science and Subs, respectively), its recall is very low (20.41 and 35.15), indicating that the model classifies many new-sense words as not-new-sense. Most of our token-level features capture the intuition that when a word token appears in new or infrequent contexts, it is likely to have gained a new sense. Our results indicate that this intuition was more fruitful for EMEA than for Science or Subs.

In contrast, the type-only features (TYPEONLY)

[4] The most frequent (*voie*) appears 3881 times; there are 60 singleton words on average across the three new domains.

[5] AUC is the probability that the classifier will assign a higher score to a randomly chosen positive example than to a randomly chosen negative example (Wikipedia, 2013).

[6] We use `http://hunch.net/~vw/` version 7.1.2, and run it with the following arguments that affect learning behavior: `--exact_adaptive_norm --power_t 0.5`

		Macro			Micro		
	AUC	P	R	F	P	R	F
EMEA							
RANDOM	50.34 ± 0.60	51.24 ± 0.59	50.09 ± 1.18	50.19 ± 0.75	47.04 ± 0.60	56.07 ± 1.99	37.27 ± 0.91
CONSTANT	50.00 ± 0.00	50.99 ± 0.00	**100.0** ± 0.00	67.09 ± 0.00	45.80 ± 0.00	**100.0** ± 0.00	**52.30** ± 0.00
TYPEONLY	55.91 ± 1.13	69.76 ± 3.45	43.13 ± 1.42	41.61 ± 1.07	**77.92** ± 2.04	50.12 ± 2.35	31.26 ± 0.63
TYPEORACLE	88.73 ± 0.00	87.32 ± 0.00	86.76 ± 0.00	87.04 ± 0.00	90.01 ± 0.00	67.46 ± 0.00	59.39 ± 0.00
TOKENONLY	78.80 ± 0.52	**69.83** ± 1.59	75.58 ± 2.61	69.40 ± 1.92	59.03 ± 1.70	62.53 ± 1.66	43.39 ± 0.94
ALLFEATURES	**79.60** ± 1.20	68.11 ± 1.19	**79.84** ± 2.27	**71.64** ± 1.83	55.28 ± 1.11	**71.50** ± 1.62	**46.83** ± 0.62
Science							
RANDOM	50.18 ± 0.78	24.48 ± 0.57	50.32 ± 1.33	32.92 ± 0.79	46.99 ± 0.51	60.32 ± 1.06	34.72 ± 1.03
CONSTANT	50.00 ± 0.00	24.34 ± 0.00	**100.0** ± 0.00	39.15 ± 0.00	44.39 ± 0.00	**100.0** ± 0.00	**50.44** ± 0.00
TYPEONLY	**77.06** ± 1.23	**66.07** ± 2.80	36.28 ± 4.10	34.50 ± 4.06	**84.97** ± 0.82	36.81 ± 2.33	24.22 ± 1.70
TYPEORACLE	88.76 ± 0.00	78.43 ± 0.00	69.29 ± 0.00	73.54 ± 0.00	84.19 ± 0.00	67.41 ± 0.00	52.67 ± 0.00
TOKENONLY	66.62 ± 0.47	60.50 ± 3.11	28.05 ± 2.06	30.81 ± 2.75	76.21 ± 1.78	36.57 ± 2.23	24.68 ± 1.36
ALLFEATURES	73.91 ± 0.66	50.59 ± 2.08	**60.60** ± 2.04	**47.54** ± 1.52	66.72 ± 1.19	**62.30** ± 1.36	**40.22** ± 1.03
Subs							
RANDOM	50.26 ± 0.69	42.47 ± 0.60	50.17 ± 0.84	45.68 ± 0.68	52.18 ± 1.32	54.63 ± 2.01	39.87 ± 2.10
CONSTANT	50.00 ± 0.00	42.51 ± 0.00	**100.0** ± 0.00	**59.37** ± 0.00	50.63 ± 0.00	**100.0** ± 0.00	**58.67** ± 0.00
TYPEONLY	67.16 ± 0.73	**76.41** ± 1.51	31.91 ± 3.15	36.37 ± 2.58	**90.03** ± 0.61	34.78 ± 1.12	26.20 ± 0.61
TYPEORACLE	81.35 ± 0.00	83.12 ± 0.00	70.23 ± 0.00	76.12 ± 0.00	90.62 ± 0.00	52.37 ± 0.00	44.43 ± 0.00
TOKENONLY	63.30 ± 0.99	63.17 ± 2.31	45.38 ± 2.07	43.30 ± 1.29	76.38 ± 1.68	49.70 ± 1.76	37.92 ± 1.20
ALLFEATURES	**69.26** ± 0.60	63.48 ± 1.77	**56.22** ± 2.66	**52.78** ± 1.96	67.55 ± 0.83	**62.18** ± 1.45	**43.85** ± 0.90

Table 4: Complete SENSESPOTTING results for all domains. The scores are from cross-validation on a single domain; in all cases, higher is better. Two standard deviations of performance over the cross-validation are shown in small type. For all domains and metrics, the highest (not necessarily statistically significant) non-oracle results are bolded.

are relatively weak for predicting new senses on EMEA data but stronger on Subs (TYPEONLY AUC performance is higher than both baselines) and even stronger on Science data (TYPEONLY AUC and f-measure performance is higher than both baselines as well as the ALLFEATURESmodel). In our experience with the three datasets, we know that the Science data, which contains abstracts from a wide variety of scientific disciplines, is the most diverse, followed by the Subs data, and then EMEA, which mostly consists of text from drug labels and tends to be quite repetitive. Thus, it makes sense that type-level features would be the most informative for the least homogeneous dataset. Representative words in scientific text are likely to appear in variety of contexts, while in the EMEA data they may only appear in a few, making it easier to contrast them with the distributions observed in the old domain data.

For all domains, in micro-level evaluation, our models fail to outperform the CONSTANT baseline. Recall that the micro-level evaluation computes precision, recall, and f-measure for all word tokens of a given word type and then averages across word types. We observe that words that are less frequent in both the old and the new domains are more likely to have a new sense than more frequent words, which causes the CONSTANT baseline to perform reasonably well. In contrast, it is more difficult for our models to make good predictions for less frequent words. A low frequency in the new domain makes type level features (estimated over only a few instances) noisy and unreliable. Similarly, a low frequency in the old domain makes the our token level features, which all contrast with old domain instances of the word type.

6.4 Feature Ablation

In the previous section, we observed that (with one exception) both Type-level and Token-level features are useful in our task (in some cases, essential). In this section, we look at finer-grained feature distinctions through a process of feature ablation. In this setting, we begin with *all features* in a model and *remove* one feature at a time, always removing the feature that hurts performance least. For these experiments, we determine which feature to remove using AUC. Note that we're actually able to beat (by 2-4 points AUC) the scores from Table 4 by removing features!

The results here are somewhat mixed. In EMEA and Science, one can actually get by (according to AUC) with very few features: just two (Type:NgramProband Type:Context) are sufficient to achieve optimal AUC scores. To get higher Macro-F scores requires nearly all the features, though this is partially due to the choice of

EMEA	AUC	MacF
AllFeatures	79.60	71.64
–Token:L-PSDBin	77.09	70.50
–Type:RelFreq	78.43	72.19
–Token:G-PSD	**79.66**	72.11
–Type:Context	**79.66**	72.45
–Token:Ctx%	78.91	**73.37**
–Type:TopicSim	78.05	71.33
–Token:CtxCnt	76.90	71.72
–Token:L-PSD	76.03	73.35
–Type:NgramProb	73.32	69.54
–Token:G-PSDBin	74.41	69.76
–Token:NgramProb	69.78	68.89
–Token:PSDRatio	48.38	3.45

Science	AUC	MacF
AllFeatures	73.91	47.54
–Token:L-PSDBin	76.26	53.69
–Token:G-PSD	77.04	53.56
–Token:G-PSDBin	77.44	54.54
–Token:L-PSD	77.85	56.05
–Token:PSDRatio	77.92	**57.34**
–Token:CtxCnt	77.85	54.42
–Type:Context	**78.17**	55.45
–Token:Ctx%	78.06	55.04
–Type:TopicSim	77.83	54.57
–Token:NgramProb	76.98	51.02
–Type:RelFreq	74.25	49.57
–Type:NgramProb	50.00	0.00

Subs	AUC	MacF
AllFeatures	69.26	52.78
–Type:NgramProb	69.13	53.33
–Token:G-PSDBin	70.23	54.72
–Token:CtxCnt	71.23	58.35
–Token:L-PSDBin	72.07	57.85
–Token:G-PSD	72.17	57.33
–Type:TopicSim	**72.31**	58.41
–Token:Ctx%	72.17	56.17
–Token:NgramProb	71.35	**59.26**
–Token:PSDRatio	70.33	46.88
–Token:L-PSD	69.05	53.31
–Type:RelFreq	65.25	48.22
–Type:Context	50.00	0.00

Table 5: Feature ablation results for all three corpora. Selection criteria is AUC, but Macro-F is presented for completeness. Feature selection is run independently on each of the three datasets. The features toward the *bottom* were the first selected.

	AUC	Macro-F	Micro-F
EMEA			
TypeOnly	71.43 ± 0.94	52.62 ± 3.41	38.67 ± 1.35
TokenOnly	**73.75** ± 1.11	**67.77** ± 4.18	45.49 ± 3.96
AllFeatures	72.19 ± 4.07	67.26 ± 7.88	**49.29** ± 3.55
XV-AllFeatures	79.60 ± 1.20	71.64 ± 1.83	46.83 ± 0.62
Science			
TypeOnly	**75.19** ± 0.89	**51.53** ± 2.55	37.14 ± 4.41
TokenOnly	71.24 ± 1.45	47.27 ± 1.11	40.48 ± 1.84
AllFeatures	74.14 ± 0.93	48.86 ± 3.94	**43.20** ± 3.16
XV-AllFeatures	73.91 ± 0.66	47.54 ± 1.52	40.22 ± 1.03
Subs			
TypeOnly	60.90 ± 1.47	39.21 ± 14.78	24.77 ± 2.78
TokenOnly	**62.00** ± 1.16	49.74 ± 6.30	**42.95** ± 3.92
AllFeatures	60.12 ± 2.11	**50.16** ± 8.63	38.56 ± 5.20
XV-AllFeatures	69.26 ± 0.60	52.78 ± 1.96	43.85 ± 0.90

Table 6: Cross-domain test results on the SenseSpotting task. Two standard deviations are shown in small type. Only AUC, Macro-F and Micro-F are shown for brevity.

AUC as the measure on which to ablate. It's quite clear that for Science, all the useful information is in the type-level features, a result that echoes what we saw in the previous section. While for EMEA and Subs, both type- and token-level features play a significant role. Considering the six most useful features in each domain, the ones that pop out as frequently most useful are the global PSD features, the ngram probability features (either type- or token-based), the relative frequency features and the context features.

6.5 Cross-Domain Training

One disadvantage to the previous method for evaluating the SenseSpotting task is that it requires parallel data in a new domain. Suppose we have *no* parallel data in the new domain at all, yet still want to attack the SenseSpotting task. One option is to train a system on domains for which we *do* have parallel data, and then apply it in a new domain. This is precisely the setting we explore in this section. Now, instead of performing cross-validation in a single domain (for instance, Science), we take the union of *all* of the training data in the other domains (e.g., EMEA and Subs), train a classifier, and then apply it to Science. This classifier will almost certainly be worse than one trained on New (Science) but does not require *any* parallel data in that domain. (Hyperparameters are chosen by development data from the Old union.)

The results of this experiment are shown in Table 6. We include results for TokenOnly, TypeOnly and AllFeatures; all of these are trained in the cross-domain setting. To ease comparison to the results that do not suffer from domain shift, we also present "XV-AllFeatures", which are results copied from Table 4 in which parallel data from New is used. Overall, there is a drop of about 7.3% absolute in AUC, moving from XV-AllFeatures to AllFeatures, including a small improvement in Science (likely because Science is markedly smaller than Subs, and "more difficult" than EMEA with many word types).

6.6 Detecting Most Frequent Sense Changes

We define a second, related task: MostFreqSenseChange. In this task, instead of predicting if a given word token has a sense which is brand new with respect to the old domain, we predict whether it is being used with a a sense which is not the one that was observed *most frequently* in the old domain. In our EMEA, Science, and Subtitles data, 68.2%, 48.3%, and 69.6% of word tokens' predominant sense changes.

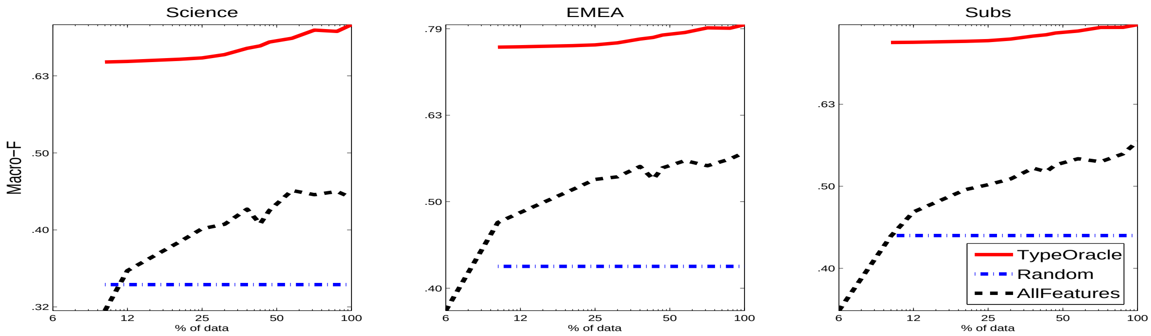

Figure 1: Learning curves for the three domains. X-axis is percent of data used, Y-axis is Macro-F score. Both axes are in log scale to show the fast rate of growth. A horizontal bar corresponding to random predictions, and the TypeOracle results are shown for comparison.

	AUC	Macro-F	Micro-F
EMEA			
Random	50.54 ± 0.41	58.23 ± 0.34	49.69 ± 0.85
Constant	50.00 ± 0.00	**82.15** ± 0.00	**74.43** ± 0.00
TypeOnly	55.05 ± 1.00	67.45 ± 1.35	65.72 ± 0.59
TypeOracle	88.36 ± 0.00	90.64 ± 0.00	77.46 ± 0.00
TokenOnly	**66.42** ± 1.07	80.27 ± 0.50	68.96 ± 0.58
AllFeatures	58.64 ± 3.45	**80.57** ± 0.45	**69.40** ± 0.51
Science			
Random	50.13 ± 0.78	49.05 ± 0.82	48.19 ± 1.47
Constant	50.00 ± 0.00	**65.21** ± 0.00	**73.22** ± 0.00
TypeOnly	68.32 ± 1.05	54.70 ± 2.35	57.04 ± 1.52
TypeOracle	91.41 ± 0.00	86.71 ± 0.00	74.26 ± 0.00
TokenOnly	**68.49** ± 0.59	62.76 ± 0.89	64.40 ± 1.08
AllFeatures	68.31 ± 0.93	**64.73** ± 1.93	**67.20** ± 1.65
Subs			
Random	50.27 ± 0.27	56.93 ± 0.29	50.93 ± 1.11
Constant	50.00 ± 0.00	**79.96** ± 0.00	**76.26** ± 0.00
TypeOnly	**60.36** ± 0.90	67.78 ± 1.98	61.58 ± 1.78
TypeOracle	82.16 ± 0.00	87.96 ± 0.00	73.87 ± 0.00
TokenOnly	59.49 ± 1.04	**77.79** ± 0.82	**73.51** ± 0.68
AllFeatures	54.97 ± 0.89	77.30 ± 1.58	72.29 ± 1.68

Table 7: Cross-validation results on the MostFreqSenseChange task. Two standard deviations are shown in small type.

We use the same set of features and learning framework to generate and evaluate models for this task. While the SenseSpotting task has MT utility in suggesting which new domain words demand a new translation, the MostFreqSenseChange task has utility in suggesting which words demand a new translation probability distribution when shifting to a new domain. Table 7 shows the results of our MostFreqSenseChange task experiments.

Results on the MostFreqSenseChange task are somewhat similar to those for the SenseSpotting task. Again, our models perform better under a macro-level evaluation than under a micro-level evaluation. However, in contrast to the SenseSpotting results, token-level features perform quite well on their own for all domains. It makes sense that our token level features have a better chance of success on this task. The important comparison now is between a new domain token in context and the *majority* of the old domain tokens of the same word type. This comparison is likely to be more informative than when we are equally interested in identifying overlap between the current token and any old domain senses. Like the SenseSpotting results, when doing a micro-level evaluation, our models do not perform as well as the Constant baseline, and, as before, we attribute this to data sparsity.

6.7 Learning Curves

All of the results presented so far use classifiers trained on instances of representative types (i.e. "representative tokens") extracted from fairly large new domain parallel corpora (see Table 3), consisting of between 22 and 36 thousand parallel sentences, which yield between 8 and 35 thousand representative tokens. Although we expect some new domain parallel tuning data to be available in most MT settings, we would like to know how many representative types are required to achieve good performance on the SenseSpotting task. Figure 6.5 shows learning curves over the number of representative tokens that are used to train SenseSpotting classifiers. In fact, only about 25-50% of the data we used is really necessary to achieve the performance observed before.

Acknowledgments We gratefully acknowledge the support of the JHU summer workshop program (and its funders), the entire DAMT team (`http://hal3.name/DAMT/`), Sanjeev Khudanpur, support from the NRC for Marine Carpuat, as well as DARPA CSSG Grant D11AP00279 for Hal Daumé III and Jagadeesh Jagarlamudi.

References

E. Agirre and P.G. Edmonds. 2006. *Word Sense Disambiguation: Algorithms and Applications*. Text, Speech, and Language Technology Series. Springer Science+Business Media B.V.

Eneko Agirre and Aitor Soroa. 2007. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 7–12.

David Bamman and Gregory Crane. 2011. Measuring historical word sense variation. In *Proceedings of the 2011 Joint International Conference on Digital Libraries (JCDL 2011)*, pages 1–10.

D. Blei, A. Ng, and M. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 3.

Michael Bloodgood and Chris Callison-Burch. 2010. Bucking the trend: Large-scale cost-focused active learning for statistical machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 854–864, Uppsala, Sweden, July. Association for Computational Linguistics.

Marine Carpuat and Dekai Wu. 2007. Improving Statistical Machine Translation using Word Sense Disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 61–72, Prague, June.

Marine Carpuat, Hal Daumé III, Alexander Fraser, Chris Quirk, Fabienne Braune, Ann Clifton, Ann Irvine, Jagadeesh Jagarlamudi, John Morgan, Majid Razmara, Aleš Tamchyna, Katharine Henry, and Rachel Rudinger. 2012. Domain adaptation in machine translation: Final report. In *2012 Johns Hopkins Summer Workshop Final Report*.

Yee Seng Chan and Hwee Tou Ng. 2007. Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the Association for Computational Linguistics*.

Paul Cook and Suzanne Stevenson. 2010. Automatically identifying changes in the semantic orientation of words. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 28–34, Valletta, Malta.

Hal Daumé III and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

Katrin Erk. 2006. Unknown word sense detection as outlier detection. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 128–135.

Pascale Fung and Lo Yuen Yee. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the google books ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 67–71, Edinburgh, UK, July. Association for Computational Linguistics.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

Matthew Hoffman, David Blei, and Francis Bach. 2010. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, Timothy Baldwin, and Lexical Computing. 2012. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for computational Linguistics (EACL 2012)*, pages 591–601. Citeseer.

Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(04):359–373.

Mausam, Stephen Soderland, Oren Etzioni, Daniel S. Weld, Kobi Reiter, Michael Skinner, Marcus Sammer, and Jeff Bilmes. 2010. Panlingual lexical translation via probabilistic inference. *Artificial Intelligence*, 174:619–637, June.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 279. Association for Computational Linguistics.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2007. Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553–590.

Behrang Mohit and Rebecca Hwa. 2007. Localization of difficult-to-translate phrases. In *proceedings of the 2nd ACL Workshop on Statistical Machine Translations*.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

Eyal Sagi, Stefan Kaufmann, and Brady Clark. 2009. Semantic density analysis: Comparing word meaning across time and phonetic space. In *Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics*, pages 104–111, Athens, Greece, March.

Charles Schafer and David Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

Charles Schafer. 2006. *Translation Discovery Using Diverse Similarity Measures*. Ph.D. thesis, Johns Hopkins University.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing (RANLP)*.

Wikipedia. 2013. Receiver operating characteristic. `http://en.wikipedia.org/wiki/Receiver_operating_characteristic#Area_Under_the_Curve`, February.