Unsupervised Discriminative Language Model Training for Machine Translation using Simulated Confusion Sets

Zhifei Li and **Ziyuan Wang** and **Sanjeev Khudanpur** and **Jason Eisner**
Center for Language and Speech Processing
Johns Hopkins University
zhifei.work@gmail.com,{zwang40, khudanpur, eisner}@jhu.edu

Abstract

An *unsupervised* discriminative training procedure is proposed for estimating a language model (LM) for machine translation (MT). An English-to-English synchronous context-free grammar is derived from a baseline MT system to capture *translation alternatives*: pairs of words, phrases or other sentence fragments that potentially compete to be the translation of the same source-language fragment. Using this grammar, a set of impostor sentences is then created for each English sentence to *simulate* confusions that would arise if the system were to process an (unavailable) input whose correct English translation is that sentence. An LM is then trained to discriminate between the original sentences and the impostors. The procedure is applied to the IWSLT Chinese-to-English translation task, and promising improvements on a state-of-the-art MT system are demonstrated.

1 Discriminative Language Modeling

A language model (LM) constitutes a crucial component in many tasks such as machine translation (MT), speech recognition, information retrieval, handwriting recognition, etc. It assigns a priori probabilities to word sequences. In general, we expect a low probability for an ungrammatical or implausible word sequence. The dominant LM used in such systems is the so-called n-gram model, which is typically derived from a large corpus of target language text via maximum likelihood estimation, mitigated by some smoothing or regularization. Due to the Markovian assumptions implicit in n-gram models, however, richer linguistic and semantic dependencies are not well captured. Rosenfeld (1996) and Khudanpur and Wu (2000) address such shortcoming by using maximum entropy models with long-span features, while still working with a *locally* normalized left-to-right LM. The whole-sentence maximum entropy LM of Rosenfeld et al. (2001) proposes a *globally* normalized log-linear LM incorporating several sentence-wide features.

The n-gram as well as the whole-sentence model are *generative* or descriptive models of text. However, in a task like Chinese-to-English MT, the de facto role of the LM is to *discriminate* among the alternative English translations being contemplated by the MT system for a particular Chinese input sentence. We call the set of such alternative translations a *confusion set*. Since a confusion set is typically a minuscule subset of the set of all possible word sequences, it is arguably better to train the LM parameters so as to make the *best candidate* in the confusion set more likely than its competitors, as done by Roark et al. (2004) for speech recognition and by Li and Khudanpur (2008) for MT. Note that identifying the best candidate requires *supervised* training data—bilingual text in case of MT—which is expensive in many domains (e.g. weblog or newsgroup) and for most language pairs (e.g. Urdu-English).

We propose a novel discriminative LM in this paper: a globally normalized log-linear LM that can be trained in an *efficient* and *unsupervised* manner, using only monolingual (English) text.

The main idea is to exploit (translation) uncertainties inherent in an MT system to derive an English-to-English confusion grammar (CG), illustrated in this paper for a Hiero system (Chiang, 2007). From the *bilingual* synchronous context-free grammar (SCFG) used in Hiero, we extract a *monolingual* SCFG, with rules of the kind, $X \to \langle \text{strong tea}, \text{powerful tea} \rangle$ or

Coling 2010: Poster Volume, pages 656–664,
Beijing, August 2010

$X \rightarrow \langle \text{in } X_1, \text{in the } X_1 \rangle$. Thus our CG is also an SCFG that generates pairs of English sentences that differ from each other in ways that alternative English hypothesis considered during translation would differ from each other. This CG is then used to "translate" each sentence in the LM training corpus into what we call its *confusion set* — a set of other "sentences" with which that sentence would likely be confused by the MT system, were it to be the target translation of a source-language sentence. Sentences in the training corpus, each paired with its confusion set, are then used to train a discriminative LM to prefer the training sentences over the alternatives in their confusion sets.

Since the monolingual CG and the bilingual Hiero grammar are both SCFGs, the confusion sets are isomorphic with translation hypergraphs that are used by supervised discriminative training. The confusion sets thus *simulate* the supervised case, with a key exception: lack of any (Chinese) source-language information. Therefore, only target-side "language model" probabilities may be estimated from confusion sets.

We carry out this discriminative training procedure, and empirically demonstrate promising improvements in translation quality.

2 Discriminative LM Training

2.1 Whole-sentence Maximum Entropy LM

We aim to train a globally normalized log-linear language model $p_\theta(y)$ of the form

$$p_\theta(y) \quad = \quad \mathrm{Z}^{-1}\, e^{f(y)\cdot\theta} \qquad (1)$$

where y is an English sentence, $f(y)$ is a vector of arbitrary features of y, θ is the (weight) vector of model parameters, and $\mathrm{Z} \overset{\text{def}}{=} \sum_{y'} e^{f(y')\cdot\theta}$ is a normalization constant. Given a set of English training sentences $\{y_i\}$, the parameters θ may be chosen to maximize likelihood, as

$$\theta^* \quad = \quad \arg\max_\theta \prod_i p_\theta(y_i). \qquad (2)$$

This is the so called whole-sentence maximum entropy (WSME) language model[1] proposed by Rosenfeld et al. (2001). Training the model of (2) requires computing Z, a sum over all possible word sequences y' with any length, which is computationally intractable. Rosenfeld et al. (2001) approximate Z by random sampling.

2.2 Supervised Discriminative LM Training

In addition to the computational disadvantage, (2) also has a modeling limitation. In particular, in a task like MT, the primary role of the LM is to discriminate among alternative translations of a *given* source-language sentence. This set of alternatives is typically a minuscule subset of all possible target-language word sequences. Therefore, a better way to train the global log-linear LM, given bilingual text $\{(x_i, y_i)\}$, is to generate the *real* confusion set $\mathcal{N}(x_i)$ for each input sentence x_i using a specific MT system, and to adjust θ to discriminate between the reference translation y_i and $y' \in \mathcal{N}(x_i)$ (Roark et al., 2004; Li and Khudanpur, 2008).

For example, one may maximize the conditional likelihood of the bilingual training data as

$$\begin{aligned} \theta^* \quad &= \quad \arg\max_\theta \prod_i p_\theta(y_i \mid x_i) \qquad (3) \\ &= \quad \arg\max_\theta \prod_i \frac{e^{f(x_i,y_i)\cdot\theta}}{\sum_{y'\in\mathcal{N}(x_i)} e^{f(x_i,y')\cdot\theta}}, \end{aligned}$$

which entails summing over *only* the candidate translations y' of the given input x_i. Furthermore, if the features $f(x_i, y)$ are depend on *only* the output y, i.e. on the English-side features of the bilingual text, the resulting discriminative model may be interpreted as a *language model*.

Finally, in a Hiero style MT system, if $f(x_i, y)$ depends on the target-side(s) of the bilingual rules used to construct y from x_i, we essentially have a *syntactic* LM.

2.3 Unsupervised Discriminative Training using Simulated Confusion Sets

While the supervised discriminative LM training has both computational and modeling advantages over the WSME LM, it relies on bilingual data, which is expensive to obtain for several domains and language pairs. For such cases, we propose a novel discriminative language model, which is

[1] Note the contrast with the maximum entropy n-gram LM (Rosenfeld, 1996; Khudanpur and Wu, 2000), where the normalization is performed for each n-gram history.

still a global log-linear LM with the modeling advantage and computational *efficiency* of (3) but requires only monolingual text $\{y_i\}$ for training θ. Specifically, we propose to modify (3) as

$$\begin{aligned}\theta^* &= \arg\max_{\theta} \prod_i p_\theta(y_i \,|\, \mathcal{N}(y_i)) \qquad (4)\\ &= \arg\max_{\theta} \prod_i \frac{e^{f(y_i)\cdot\theta}}{\sum_{y' \in \mathcal{N}(y_i)} e^{f(y')\cdot\theta}},\end{aligned}$$

where $\mathcal{N}(y_i)$ is a *simulated* confusion set for y_i obtained by applying a confusion grammar to y_i, as detailed in Section 3. Our hope is that $\mathcal{N}(y_i)$ resembles the actual confusion set $\mathcal{N}(x_i)$ that an MT system would generate if it were given the input sentence x_i.

Like (3), the maximum likelihood training of (4) does not entail the expensive computation of a global normalization constant Z, and is therefore very *efficient*. Unlike (3) however, where the input x_i for each output y_i is needed to create $\mathcal{N}(x_i)$, the model of (4) can be trained in an *unsupervised* manner with only $\{y_i\}$.

3 Unsupervised Discriminative Training of the Language Model for MT

The following is thus the proposed procedure for unsupervised discriminative training of the LM.

1. Extract a *confusion grammar* (CG) from the baseline MT system.
2. "Translate" each English sentence in the LM training corpus, using the CG as an English-to-English translation model, to generate a *simulated* confusion set.
3. Train a discriminative language model on the simulated confusion sets, using the corresponding original English sentences as the training references.

The trained model may then be used for actual MT decoding. We next describe each step in detail.

3.1 Extracting a Confusion Grammar

We assume a synchronous context free grammar (SCFG) formalism for the confusion grammar (CG). While the SCFG used by the MT system is bilingual, the CG we extract will be *monolingual*, with both the source and target sides being English. Some example CG rules are:

$$\begin{aligned} X &\rightarrow \langle\, \text{strong tea}\,, \text{powerful tea}\,\rangle\,,\\ X &\rightarrow \langle\, X_0 \text{ at beijing}\,, \text{beijing 's } X_0\,\rangle\,,\\ X &\rightarrow \langle\, X_0 \text{ of } X_1\,,\ X_0 \text{ of the } X_1\,\rangle\,,\\ X &\rightarrow \langle\, X_0 \text{ 's } X_1\,,\ X_1 \text{ of } X_0\,\rangle\,.\end{aligned}$$

Like a regular SCFG, a CG contains rules with different "arities" and reordering of the nonterminals (as shown in the last example) capturing the confusions that the MT system encounters when choosing word *senses*, *reordering patterns*, etc.

3.1.1 Extracting a Confusion Grammar from the Bilingual Grammar

The confusion grammar is derived from the MT system's bilingual grammar. In Hiero, the bilingual rules are of the form $X \rightarrow \langle c, e\rangle$, where both c and e may contain (a matched number of) nonterminal symbols. For every c which appears on the source-side of two different Hiero rules $X \rightarrow \langle c, e_1\rangle$ and $X \rightarrow \langle c, e_2\rangle$, we extract two CG rules, $X \rightarrow \langle e_1, e_2\rangle$ and $X \rightarrow \langle e_2, e_1\rangle$, to capture the confusion the MT system would face were it to encounter c in its input. For each Hiero rule $X \rightarrow \langle c, e\rangle$, we also extract $X \rightarrow \langle e, e\rangle$, the *identity* rule. Therefore, if a pattern c appears with $|E|$ different translation options, we extract $|E|^2$ different CG rules from c. In our current work, the rules of the CG are unweighted.

3.1.2 Test-set Specific Confusion Grammars

If the bilingual grammar contains all the rules that are extractable from the bilingual training corpus, the resulting confusion grammar is likely to be huge. As a way of reducing computation, the bilingual grammar can be restricted to a specific test set, and only rules used by the MT system for translating the test set used for extracting the CG.[2]

To economize further, one may extract a CG from the translation *hypergraphs* that are generated for the test-set. Recall that a *node* in a hypergraph corresponds to a specific source (Chinese) span, and the node has many incident *hyperedges*, each associated with a different bilin-

[2] Test-set specific CGs are of course only practical for offline applications.

gual rule. Therefore, all the bilingual rules associated with the incoming hyperedges of a given node translate the same Chinese string. At each hypergraph node, we extract CG rules to represent the competing English sides as described above. Note that even though different rules associated with a node may have different "arity," we extract CG rules only from pairs of bilingual rules that have the same arity.

A CG extracted from only the bilingual rule pairs incident on the same node in the test hypergraphs is, of course, much smaller than a CG extracted from the entire bilingual grammar. It is also more suitable for our task, since the test hypergraphs have already benefited from a baseline n-gram LM and pruning, removing all confusions that are easily resolved (rightly or wrongly) by other system components.

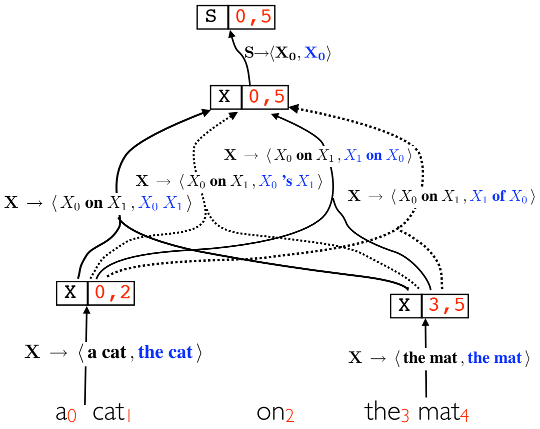
3.2 Generating Simulated Confusion Sets

For each English sentence y in the training corpus, we use the extracted CG to produce a simulated confusion set $\mathcal{N}(y)$. This is done like a regular MT decoding pass, because we can treat the CG as a Hiero style "translation" grammar[3] for an English-to-English translation system.

Since the CG is an SCFG, the confusion set $\mathcal{N}(y)$ generated for a sentence y is a *hypergraph*, encoding not only the alternative sentences y' but also the hierarchical derivation tree for each y' from y (e.g., which phrase in y has been replaced with what in y'). As usual, many different derivation trees d may correspond to the same string/sentence y' due to spurious ambiguity. We use $\mathrm{D}(y)$ to denote the set of derivations d, which is a hypergraph representation of $\mathcal{N}(y)$.

$$\begin{array}{lcl}
X & \rightarrow & \langle \text{a cat} \,,\ \text{the cat} \rangle \\
X & \rightarrow & \langle \text{the mat} \,,\ \text{the mat} \rangle \\
X & \rightarrow & \langle X_0 \text{ on } X_1 \,,\ X_0\ X_1 \rangle \\
X & \rightarrow & \langle X_0 \text{ on } X_1 \,,\ X_0 \text{ 's } X_1 \rangle \\
X & \rightarrow & \langle X_0 \text{ on } X_1 \,,\ X_1 \text{ on } X_0 \rangle \\
X & \rightarrow & \langle X_0 \text{ on } X_1 \,,\ X_1 \text{ of } X_0 \rangle \\
S & \rightarrow & \langle X_0 \,,\ X_0 \rangle
\end{array}$$

(a) An example confusion grammar.



(b) An example hypergraph generated by the confusion grammar of (a) for the input sentence "*a cat on the mat*."

Figure 1: **Example confusion grammar and simulated confusion hypergraph.** Given an input sentence y = "*a cat on the mat*," the confusion grammar of (a) generates a hypergraph $\mathrm{D}(y)$ shown in (b), which represents the confusion set $\mathcal{N}(y)$ containing four alternative sentences y'.

Figure 1 presents an example confusion hypergraph for the English sentence y ="*a cat on the mat*," containing four alternative hypotheses: $\mathcal{N}(y)$ = { "*the cat the mat*," "*the cat 's the mat*," "*the mat of the cat*," "*the mat on the cat*"}.

Notice that each competitor $y' \in \mathcal{N}(y)$ can be regarded as the result of a "round-trip" translation $y \rightarrow x \rightarrow y'$, in which we reconstruct a possible Chinese source sentence x that our Hiero bilingual grammar could translate into both y and y'.[4] We will train our LM to prefer y, which was actually observed. Our CG-based round-trip forces $x \rightarrow y'$ to use the *same* hierarchical segmentation of x as $y \rightarrow x$ did. This constraint leads to efficient training but artificially reduces the diversity

[3] To make sure that we produce at least one derivation tree for each y, we need to add to the CG the following two glue rules, as done in Hiero (Chiang, 2007).

$$\begin{array}{lcl}
S & \rightarrow & \langle X_0 \,, X_0 \rangle \,, \\
S & \rightarrow & \langle S_0\ X_1 \,, S_0\ X_1 \rangle \,.
\end{array}$$

We also add an out of vocabulary rule $X \rightarrow \langle word, \mathrm{oov} \rangle$ for each *word* in y and set the cost of this rule to a high value so that the OOV rule will get used only when the CG does not know how to "translate" the *word*.

[4] This is because of the way we construct our CG from the Hiero grammar. However, the identity and glue rules in our CG allow almost any portion of y to be preserved untranslated through the entire $y \rightarrow x \rightarrow y'$ process. Much of y will necessarily be preserved in the situation where the CG is extracted from a small test set and hence has few non-identity rules. See (Li, 2010) for further discussion.

of $\mathcal{N}(y)$. In other recent work (Li et al., 2010), we have taken the round-trip view more seriously, by imputing *likely* source sentences x and translating them back to *separate*, *weighted* confusion forests $\mathcal{N}(y)$, *without* any same-segmentation constraint.

3.3 Confusion-based Discriminative Training

With the training sentences y_i and their simulated confusion sets $\mathcal{N}(y_i)$ — represented as hypergraphs $\mathrm{D}(y_i)$) — we can perform the discriminative training using any of a number of procedures such as MERT (Och, 2003) or MIRA as used by Chiang et al. (2009). In our paper, we use hypergraph-based minimum risk (Li and Eisner, 2009),

$$\begin{aligned}\theta^* &= \arg\min_{\theta} \sum_i \mathrm{Risk}_\theta(y_i) \qquad (5)\\ &= \arg\min_{\theta} \sum_i \sum_{d \in \mathrm{D}(y_i)} \mathrm{L}(\mathrm{Y}(d), y_i) p_\theta(d \mid \mathrm{D}(y_i)),\end{aligned}$$

where $\mathrm{L}(y', y_i)$ is the loss (e.g negated BLEU) incurred by producing y' when the true answer is y_i, $\mathrm{Y}(d)$ is the English *yield* of a derivation d, and $p_\theta(d \mid \mathrm{D}(y_i))$ is defined as,

$$p_\theta(d \mid \mathrm{D}(y_i)) = \frac{e^{f(d)\cdot\theta}}{\sum_{d \in \mathrm{D}(y_i)} e^{f(d)\cdot\theta}}, \qquad (6)$$

where $f(d)$ is a feature vector over d. We will specify the features in Section 5, but in general they should be defined such that the training will be efficient and the actual MT decoding can use them conveniently.

The objective of (5) is differentiable and thus we can optimize θ by a gradient-based method. The risk and its gradient on a hypergraph can be computed by using a second-order expectation semiring (Li and Eisner, 2009).

3.3.1 Iterative Training

In practice, the full confusion set $\mathcal{N}(y)$ defined by a confusion grammar may be too large and we have to perform pruning when training our model. But the pruning itself may depend on the model that we aim to train. How do we solve this circular dependency problem? We adopt the following procedure. Given an initial model θ, we generate a hypergraph (with pruning) for each y, and train an optimal θ^* of (5) on these hypergraphs. Then, we use the optimal θ^* to *regenerate* a hypergraph for each y, and do the training again. This iterates until convergence. This procedure is quite similar to the k-best MERT (Och, 2003) where the training involves a few iterations, and each iteration uses a new k-best list generated using the latest model.

3.4 Applying the Discriminative LM

First, we measure the goodness of our language model in a simulated task. We generate simulated confusion sets $\mathcal{N}(y)$ for some held out English sentences y, and test how well $p_\theta(d \mid \mathrm{D}(y))$ can recover y from $\mathcal{N}(y)$. This is merely a proof of concept, and may be useful in deciding which features $f(d)$ to employ for discriminative training.

The intended use of our model is, of course, for actual MT decoding (e.g., translating Chinese to English). Specifically, we can add the discriminative model into an MT pipeline as a feature, and tune its weight relative to other models in the MT system, including the baseline n-gram LM.

4 Related and Similar Work

The detailed relation between the proposed procedure and other language modeling techniques has been discussed in Sections 1 and 2. Here, we review two other methods that are related to our method in a broader context.

4.1 Unsupervised Training of Global Log-linear Models

Our method is similar to the contrastive estimation (CE) of Smith and Eisner (2005) and its successors (Poon et al., 2009). In particular, our confusion grammar is like a *neighborhood function* in CE. Also, our goal is to improve both efficiency and accuracy, just as CE does. However, there are two important differences. First, the neighborhood function in CE is manually created based on human insights about the particular task, while our neighborhood function, generated by the CG, is automatically *learnt* (e.g., from the bilingual grammar) and specific to the MT system being used. Therefore, our neighborhood function is more likely to be informative and adaptive to the task. Secondly, when tuning θ, CE uses the maximum likelihood training, but we use the minimum risk training of (5). Since our training uses a task-specific loss function, it is likely to perform better than maximum likelihood training.

4.2 Paraphrasing Models

Our method is also related to methods for training paraphrasing models (Quirk et al., 2004; Bannard and Callison-Burch, 2005; Callison-Burch et al., 2006; Madnani et al., 2007). Specifically, the form of our *confusion grammar* is similar to that of the *paraphrase model* they use, and the ways of extracting the grammar/model are also similar as both employ a second language (e.g., Chinese in our case) as a *pivot*. However, while a "translation" rule in a paraphrase model is expected to contain a pair of phrases that are good alternatives for each other, a confusion rule in our CG is based on an MT system processing unseen test data and contains pairs of phrases that are typically bad (and only rarely good) alternatives for each other.

The motivation and goal are also different. For example, the goal of Bannard and Callison-Burch (2005) is to extract paraphrases with the help of parallel corpora. Callison-Burch et al. (2006) aim to improve MT quality by adding paraphrases in the translation table, while Madnani et al. (2007) aim to improve the minimum error rate training by adding the automatically generated paraphrases into the English reference sets. In contrast, our motivation is to train a *discriminative* language model to improve MT (by using the confusion grammar to decide what alternatives the model should learn to discriminate).

5 Experimental Results

We have applied the confusion-based discriminative language model (CDLM) to the IWSLT 2005 Chinese-to-English text translation task[5] (Eck and Hori, 2005). We see promising improvements over an n-gram LM for a solid **Joshua**-based baseline system (Li et al., 2009).

5.1 Data Partitions for Training & Testing

Four kinds of data are used for CDLM training:

	Data Usage	# sentences	
		ZH	EN
Set1	TM & LM training	40k	40k
Set2	Min-risk training	1006	1006×16
Set3	CDLM training	—	1006×16
Set4	Test	506	506×16

Table 1: **Data sets used.** Set1 contains translation-equivalent Chinese-English sentence pairs, while for each Chinese sentence in Set2 and Set4, there are 16 English translations. Set3 happens to be the English side of Set2 due to lack of additional in-domain English text, but this is not noteworthy; Set3 could be any in-domain target-language text corpus.

Set1 a bilingual training set on which 10 individual MT system components are trained,

Set2 a small bilingual, in-domain set for tuning relative weights of the system components,

Set3 an in-domain monolingual target-language corpus for CDLM training, and

Set4 a test set on which improvements in MT performance is measured.

We partition the IWSLT data into four such subsets as listed in Table 1.

5.2 Baseline MT System

Our baseline translation model components are estimated from 40k pairs of utterances from the travel domain, called Set1 in Table 1. We use a *5-gram* language model with modified Kneser-Ney smoothing (Chen and Goodman, 1998), trained on the English side of Set1, as our baseline LM.

The baseline MT system comprises 10 component models (or "features") that are standard in Hiero (Chiang, 2007), namely the baseline language model (BLM) feature, three baseline translation model features, one word-insertion penalty (WP) feature, and five *arity* features — three to count how many rules with an arity of zero/one/two are used in a derivation, and two to count how many times the unary and binary glue rules are used in a derivation. The relative weights of these 10 features are tuned via hypergraph-based minimum risk training (Li and Eisner, 2009) on the *bilingual* data Set2.

The resulting MT system gives a BLEU score of 48.5% on Set4, which is arguably a solid baseline.

[5] This is a relatively small task compared to, say, the NIST MT tasks. We worked on it for a proof-of-concept. Having been successful, we are now investigating larger MT tasks.

5.3 Unsupervised Training of the CDLM

We extract a test-set specific CG from the hypergraphs obtained by decoding Set2 and Set4, as described in Section 3.1.2. The number of rules in the bilingual grammar and the CG are about 167k and 1583k respectively. The CG is used as the "translation" model to generate confusion hypergraphs for sentences in Set3.

Two CDLMs, corresponding to different feature sets $f(d)$ in equation (6), were trained.

Only n-gram LM Features: We consider a CDLM with only two features $f(d)$: a baseline LM feature (BLM) that equals the 5-gram probability of $\mathrm{Y}(d)$ and a word penalty feature (WP) equal to the length of $\mathrm{Y}(d)$.

Target-side Rule Bigram Features[6]: For each CG rule used in d, we extract counts of bigrams that appear on the target-side of the CG rule. For example, if the confusion rule $X \rightarrow \langle\, X_0$ of X_1 , X_0 of the $X_1 \,\rangle$ is used in d, the bigram features in $f(d)$ whose counts are incremented are: "*X of*," "*of the*" and "*the X*."[7] Note that the indices on the nonterminals in the rule have been removed. To avoid very rare features, we only consider the 250 most freqent terminal symbol (English words) in the English of Set1 and map all other terminal symbols into a single class. Finally, we replace the identities of words with their dominant POS tags. These restrictions result in 525 target-side rule bigram (TsRB) features $f(d)$ in the model of (6).

For each choice of the feature vector $f(d)$, be it 2- or 527-dimensional, we use the training procedure of Section 3.3.1 to iteratively minimize the objective of (5) and get the CDLM parameter θ^*.

Note that each English sentence in Set3 has 15 other paraphrases. We generate a separate confusion hypergraph $\mathrm{D}(y)$ for each English sentence y, but for each such hypergraph we use both y and its 15 paraphrases as "reference translations" when computing the risk $\mathrm{L}(\mathrm{Y}(d), \{y\})$ in (5).[8]

5.4 Results on Monolingual Simulation

We first probe how our novel CDLM performs as a language model itself. One usually uses the perplexity of the LM on some unseen text to measure its goodness. But since we did not optimize the CDLM for likelihood, we instead examine how it performs in discriminating between a good English sentence and sentences with which the MT system may confuse that sentence. The test is performed as follows. For each test English sentence y of Set4, the confusion grammar defines a full confusion set $\mathcal{N}(y)$ via a hypergraph $\mathrm{D}(y)$. We use a LM to pick the most likely y^* from $\mathcal{N}(y)$, and then compute its BLEU score by using y and its 15 paraphrase sentences as references. The higher the BLEU, the better is the LM in picking out a good translation from $\mathcal{N}(y)$.

Table 2 shows the results[9] under a regular n-gram LM and the two CDLMs described in Section 5.3.

The baseline LM (BLM) entails no weight optimization a la (5) on Set3. The CDLM with the BLM and word pentaly (WP) features improves over the baseline LM. Compared to either of them, the CDLM with the target-side rule bigram features (TsRB) performs dramatically better.

5.5 Results on MT Test Data

We now examine how our CDLM performs during actual MT decoding. To incorporate the CDLM into MT decoding, we add the log-probability (6) of a derivation d under the CDLM as an additional

[6] Note that these features are novel in MT.

[7] With these target-side rule-based features, our LM is essentially a *syntactic* LM, not just an LM on English strings.

[8] We take unfair advantage of this unusual dataset to combat an unrelated complication—a seemingly problematic instability in the minimum risk training procedure.

As an illustration of this problem, we note that in supervised tuning of the baseline MT system ($|f(d)|$=10) with 500 sentences from Set2, the BLEU score on Set4 varies from 38.6% to 44.2% to 47.8% if we use 1, 4 and 16 reference translations during the supervised training respectively. We choose a system tuned on 16 references on Set2 as our baseline. In order not to let the unsupervised CDLM training suffer from this unrelated limitation of the tuning procedure, we give it too the benefit of being able to compute risk on Set3 using y plus its 15 paraphrases.

We wish to emphasize that this trait of Set3 having 15 paraphrases for each sentence is otherwise unnecessary, and *does not* detract much from the main claim of this paper.

[9] Note that the scores in Table 2 are very low compared to scores for actual translation from Chinese shown in Table 3. This is mainly because in this monolingual simulation, the LM is the only model used to rank the $y' \in \mathcal{N}(y)$. Said differently, y^* is being chosen in Table 2 entirely for its fluency with no consideration whatsoever for its adequacy.

LM used for	Features used			BLEU
rescoring	BLM	WP	TsRB	on Set4
Baseline LM	✓			12.8
CDLM	✓	✓		14.2
CDLM	✓	✓	✓	25.3

Table 2: **BLEU scores in monolingual simulations.** Rescoring the *confusion sets* of English sentences created using the CG shows that the CDLM with TsRB features recovers hypotheses much closer to the sentence that generated the confusion set than does the baseline n-gram LM.

Model used	Features used		BLEU
for rescoring	10 models	TsRB	on Set4
Joshua	✓		48.5
+ CDLM	✓	✓	49.5

Table 3: **BLEU scores on the test set.** The baseline MT system has ten models/features, and the proposed system has one *additional* model, the CDLM. Note that for the CDLM, only the TsRB features are used during MT decoding.

feature, on top of the 10 features already present in baseline MT system (see Section 5.2). We then (re)tune relative weights for these 11 features on the *bilingual* data Set2 of Table 1.

Note that the MT system also uses the BLM and WP features whose weights are now retuned on Set2. Therefore, when integrating a CDLM into MT decoding, it is mathematically equivalent to use only the TsRB features of the CDLM, with the corresponding weights as estimated alongside its "own" BLM and WP features during unsupervised discriminative training on Set3.

Table 3 reports the results. A BLEU score improvement of 1% is seen, reinforcing the claim that the unsupervised CDLM helps select better translations from among the system's alternatives.

5.6 Goodness of Simulated Confusion Sets

The confusion set $\mathcal{N}(y)$ generated by applying the CG to an English sentence y aims to simulate the real confusion set that would be generated by the MT system if the system's input was the Chinese sentence whose English translation is y. We investigate, in closing, how much the simulated confusion set resembles to the real one. Since we know the actual input-output pairs (x_i, y_i) for Set4, we generate two confusion sets: the simulated set $\mathcal{N}(y_i)$ and the real one $\mathcal{N}(x_i)$.

One way to measure the goodness of $\mathcal{N}(y_i)$ as a proxy for $\mathcal{N}(x_i)$, is to extract the n-gram types witnessed in the two sets, and compute the ratio of the number of n-grams in the intersection to the number in their union. Another is to measure the precision and recall of $\mathcal{N}(y_i)$ relative to $\mathcal{N}(x_i)$.

n-gram	Precision	Recall
unigram	36.5%	48.2%
bigram	10.1%	12.8%
trigram	3.7%	4.6%
4-gram	2.0%	2.4%

Table 4: **n-gram precision and recall of simulated confusion sets** relative to the true confusions when translating Chinese sentences. The n-grams are collected from k-best strings in both cases, with $k = 100$. The precision and recall change little when varying k.

Table 4 presents such precision and recall figures. For convenience, the n-grams are collected from the 100-best strings, instead of the hypergraph $\mathrm{D}(y_i)$ and $\mathrm{D}(x_i)$. Observe that the simulated confusion set does a reasonably good job on the real unigram confusions but the simulation needs improving for higher order n-grams.

6 Conclusions

We proposed a novel procedure to discriminatively train a globally normalized log-linear language model for MT, in an efficient and unsupervised manner. Our method relies on the construction of a confusion grammar, an English-to-English SCFG that captures translation alternatives that an MT system may face when choosing a translation for a given input. For each English training sentence, we use this confusion grammar to generate a simulated confusion set, from which we train a discriminative language model that will prefer the original English sentence over sentences in the confusion set. Our experiments show that the novel CDLM picks better alternatives than a regular n-gram LM from simulated confusion sets, and improves performance in a real Chinese-to-English translation task.

7 Acknowledgements

This work was partially supported by the National Science Foundation via grants № SGER-0840112 and RI-0963898, and by the DARPA GALE program. The authors thank Brian Roark and Damianos Karakos for insightful discussions.

References

Bannard, Colin and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604, Morristown, NJ, USA. Association for Computational Linguistics.

Callison-Burch, Chris, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24, Morristown, NJ, USA. Association for Computational Linguistics.

Chen, Stanley F. and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report.

Chiang, David, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *NAACL*, pages 218–226.

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Eck, Matthias and Chiori Hori. 2005. Overview of the iwslt 2005 evaluation campaign. In *In Proc. of the International Workshop on Spoken Language Translation*.

Khudanpur, Sanjeev and Jun Wu. 2000. Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling. In *Computer Speech and Language*, number 4, pages 355–372.

Li, Zhifei and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 40–51, Singapore, August. Association for Computational Linguistics.

Li, Zhifei and Sanjeev Khudanpur. 2008. Large-scale discriminative n-gram language models for statistical machine translation. In *AMTA*, pages 133–142.

Li, Zhifei, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar. Zaidan. 2009. Joshua: An open source toolkit for parsing-based machine translation. In *WMT09*, pages 26–30.

Li, Zhifei, Ziyuan Wang, Jason Eisner, and Sanjeev Khudanpur. 2010. Minimum imputed risk training for machine translation. In review.

Li, Zhifei. 2010. Discriminative training and variational decoding in machine translation via novel algorithms for weighted hypergraphs. PHD Dissertation, Johns Hopkins University.

Madnani, Nitin, Necip Fazil Ayan, Philip Resnik, and Bonnie J. Dorr. 2007. Using paraphrases for parameter tuning in statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, Prague, Czech Republic, June. Association for Computational Linguistics.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167.

Poon, Hoifung, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217, Morristown, NJ, USA. Association for Computational Linguistics.

Quirk, Chris, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142–149.

Roark, Brian, Murat Saraclar, Michael Collins, and Mark Johnson. 2004. Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 47–54, Barcelona, Spain, July.

Rosenfeld, Roni, Stanley F. Chen, and Xiaojin Zhu. 2001. Whole-sentence exponential language models: a vehicle for linguistic-statistical integration. *Computers Speech and Language*, 15(1).

Rosenfeld, Roni. 1996. A maximum entropy approach to adaptive statistical language modeling. In *Computer Speech and Language*, number 3, pages 187–228.

Smith, Noah A. and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the Association for Computational Linguistics (ACL 2005)*, Ann Arbor, Michigan.