Transfer Learning for Bilingual Content Classification*

Qian Sun[1], Mohammad S. Amin[2], Baoshi Yan[2], Craig Martell[2],
Vita Markman[2], Anmol Bhasin[3], and Jieping Ye[4]
[1]Arizona State University, qsun21@asu.edu
[2]LinkedIn Corporation, {mamin, byan, cmartell, vmarkman}@linkedin.com
[3]Groupon, Inc., abhasin@groupon.com
[4]University of Michigan, jpye@umich.edu

ABSTRACT

LinkedIn Groups provide a platform on which professionals with similar background, target and specialities can share content, take part in discussions and establish opinions on industry topics. As in most online social communities, spam content in LinkedIn Groups poses great challenges to the user experience and could eventually lead to substantial loss of active users. Building an intelligent and scalable spam detection system is highly desirable but faces difficulties such as lack of labeled training data, particularly for languages other than English. In this paper, we take the spam (Spanish) job posting detection as the target problem and build a generic machine learning pipeline for multi-lingual spam detection. The main components are feature generation and knowledge migration via transfer learning. Specifically, in the feature generation phase, a relatively large labeled data set is generated via machine translation. Together with a large set of unlabeled human written Spanish data, unigram features are generated based on the frequency. In the second phase, machine translated data are properly reweighted to capture the discrepancy from human written ones and classifiers can be built on top of them. To make effective use of a small portion of labeled data available in human written Spanish, an adaptive transfer learning algorithm is proposed to further improve the performance. We evaluate the proposed method on LinkedIn's production data and the promising results verify the efficacy of our proposed algorithm. The pipeline is ready for production.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications-Data Mining

General Terms

Algorithm

*Major part of this work was done when Qian Sun interned at LinkedIn.

KDD'15, August 11-14, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2788575.

Keywords

transfer learning, classification, text mining, NLP

1. INTRODUCTION

Spam content posted in group discussion at LinkedIn will jeopardize other user's return visit on the site. Figure 1 is one of the spam examples in group discussion[1], where A.A puts a job post in reply to S.S's profile update. A.A's response would probably annoy S.S since her reply is not related to S.S's update. Simply blocking A.A's account is not a smart solution for LinkedIn, as she may be a legitimate user and may also put informative posts somewhere else on the site. A better solution is to examine each piece of content and filter those deemed to be spammy. With the development of globalization in social network, LinkedIn is available in more than 20 languages. Therefore such spam content may be written in different languages, such as English, Spanish, Chinese, etc. Efficient tools for spam detection in multiple languages are highly desired at LinkedIn. The multilingual text categorization problem has gained its attention recently [1, 13, 15, 23, 26].

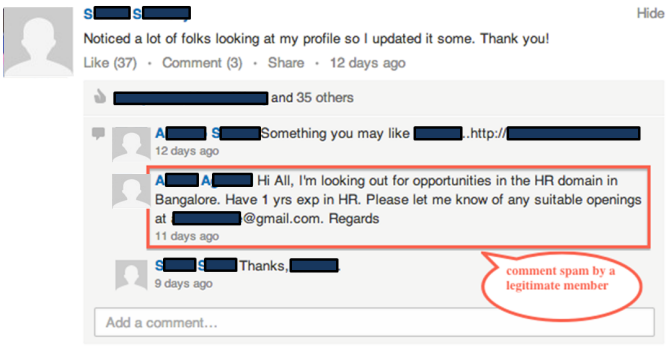


Figure 1: A sample of spam. S.S posted a status update at LinkedIn. A.A replied with a post searching for jobs, which is clearly a spam in this case.

One may solve this multi-lingual spam detection problem by enriching data representation with machine translated features in different languages [2, 21], as illustrated in Figure 2. If we are dealing with multilingual spam detection problem in English, Spanish and French, we can

[1]For the purpose of privacy protection, we anonymize the name and profile pictures in all examples.

translate any content into the other two languages. In this way, we will have bag-of-words features concatenated from three languages, and each piece of content needs to be translated into multiple languages in order to apply the uniform model. However, the machine translation to multiple languages leads to higher latency and consumes more budget. Therefore, the uniform model is not the best choice for an online product.

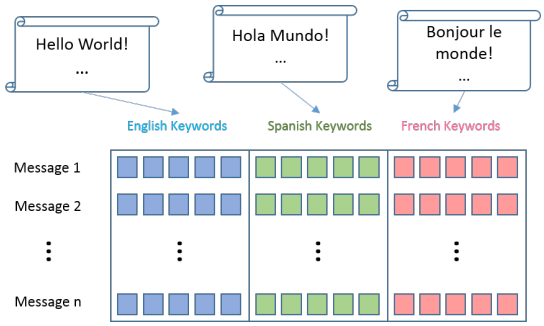


Figure 2: Multi-lingual uniform model. Messages are translated into other languages at first. The keywords for all the languages are then extracted. Frequency based values are assigned to each entry of the data matrix.

Considering the problem from an online product view, we propose to build a generic pipeline for each language. Thus we will keep a model pool in our system which contains models for different languages. The main idea of the pipeline is illustrated in Figure 3. When a piece of content is posted and transferred to the server, we first detect the language in which it is written, then apply the corresponding model to do the spam detection. If it is a spam, the system takes appropriate action. We will start with Spanish spam detection first, as Spanish is the second most widely used language in the US. The pipeline can be petentially well generalized to many other languages, such as French, German, etc.

Problem Setup: In this paper, we aim to develop a framework to deal with the Spanish spam detection problem without training data at LinkedIn. The challenge of Spanish spam detection is that we do not have enough labeled Spanish data in our database, so we cannot train an accurate model from real Spanish data. Instead we have a lot of labeled English content available. Asking human experts to help us label the Spanish data is both time and labor consuming. It is more efficient to translate Spanish content with the emergence of machine translation tools [19, 7].

There are two ways to do the translation: the first one is to translate the features (unigram) in English to Spanish, and the second one is to translate the whole content from English to Spanish and then generate the unigram features based on the machine translated Spanish. To simply translate the features in English to Spanish will lose the intended meaning of the unigrams due to the loss of the context surrounding them. For example, **Python** can be translated into one kind of programming language, and it can also be translated into snake, depending on its surrounding text. Furthermore, for certain languages there is no one-to-one correspondence in translation. For example, one unigram feature in English may be translated into several characters in Chinese. Therefore, we will follow the second way to do the translation.

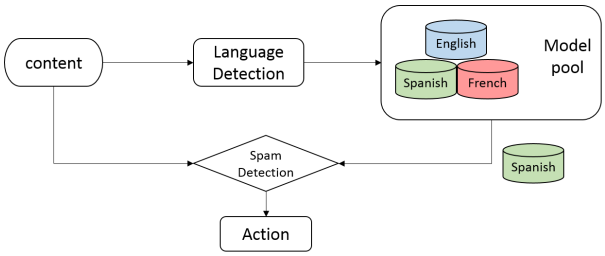


Figure 3: High-level idea for multilingual spam detection. Given a piece of content, we first detect which language it is written in. The corresponding model is then extracted from the model pool to do spam detection. Action will be taken thereafter.

Contributions: In this paper, we propose a generic pipeline to solve the multi-lingual spam detection problem. Job posting is the most common category of spam under group discussion, therefore we target at the job classification here. Starting with Spanish, we utilize the machine translation to obtain the training content in the target language. Google translator is applied in translating the labeled English content into Spanish content. We also collected job posts in Spanish that were originally written by human for model evaluation. It can be shown that the machine translated Spanish and the human written Spanish have different distributions. We therefore propose a two-stage transfer learning algorithm to migrate the knowledge learnt from machine translated Spanish to human written Spanish. In the first stage, we apply the Kernel Mean Matching (KMM) to reweight the instances in the machine translated Spanish to match the human written Spanish. In the second stage, we adaptively update the model if a small amount of label information is available in the human written Spanish domain. Our experiments on LinkedIn's production data show that the proposed two-step transfer learning algorithm achieves much better performance in terms of precision and recall compared with baseline algorithms.

Organization: The rest of the paper is organized as follows. In Section 2, we briefly review the spam detection problem in group discussion module at LinkedIn. In Section 3, we introduce our generic pipeline to solve multi-lingual spam detection problems in detail. Next, we propose our two-step transfer learning algorithm for the bilingual spam detection problem in Section 4, which consists of an instance re-weighting algorithm as step one and an adaptive learning approach as step two. In Section 5, we evaluate our proposed algorithm on LinkedIn's production data. Section 6 discusses some related work and the differences from ours. Finally, we conclude the paper in Section 7.

2. BACKGROUND

LinkedIn is the world's largest professional network with more than 332 million members in over 200 countries and territories. LinkedIn provides a wide variety of products and services that allow members to connect with one another in order to be more productive and successful in their professional lives. LinkedIn Groups in particular is a product which allows professionals in the same industry or with

similar interests to gather, share content, find answers to questions, post and view jobs, make business contacts, and establish themselves as industry experts. LinkedIn members are able to freely join public groups, or may be invited to private groups by a group manager. So far, there are more than 2 million groups at LinkedIn, and 154,000 pieces of content are being posted every day. Members who use Groups are 5 times more likely to get profile views from people outside their network. Figure 4 shows a LinkedIn group for professional interior designers, and a snapshot of discussions amongst designers sharing expertise and knowledge about their industry. The best way to participate in LinkedIn groups in general is to focus on intelligent, meaningful posts that add value to a discussion and that will benefit other members.

There are rules for participating in LinkedIn Groups discussions. First, different groups have different norms and expectations around discussions, established by the participants in the group and the group manager(s). For example, different groups have different tolerance for promotional content in group discussions. Some groups encourage members to freely share links to their own websites, blog posts, or other promotional materials, whereas other groups ask their members to confine this sort of content and sharing to a Promotions area within the group. Some groups provide a Jobs area for posting job openings, talking about the current job market, asking job related questions, or starting career discussions. Other groups allow neither promotions nor job postings. In other words, users posting to a group should understand the community norms and expectations for that group, and use good judgement when making a post. Content posted to a group that does not adhere to the group's norms and expectations is considered spam.

The motivation of this work is to develop a computational method to automatically maintain the order of group discussion, i.e., to detect and filter the unwanted posts. For each group, spam detection is conducted based on its specific rules. For example, in the groups which do not allow job posting, any posts related to job annoucement are considered as spam and appropriate actions need to be taken. Users who post spam content under group discussion are not necessarily spamers, instead they may be legitimate users but unfamiliar with the rules of the group. For this reason, we choose to build a content classifier rather than to block the users' accounts.

Users can post their updates in up to 20 different languages (Figure 5). Thus, to maintain the order of group discussion, spam detection is required to work in different languages. From the perspective of production, we build one content classifier for each language and keep a classifier pool in our system.

In summary, the key contributions of this paper are as follows: (1) We design a generic pipeline for spam detection in different languages. (2) Starting with Spanish, we develop a two-step adaptive learning approach to solve the cross-language spam detection problem. (3) We test and validate the proposed algorithm on real data from LinkedIn. The resulting pipeline is production ready.

3. PROPOSED PIPELINE

In this section, we introduce the proposed pipeline to tackle the multi-lingual spam detection problem at LinkedIn. Specifically, we design a generic pipeline to do the spam detection for different languages. Here, we start with the spam detection in Spanish, as Spanish is the second most widely used language in the US. The pipeline can be easily generated to any other language than Spanish by substituting the target language of the machine translator. The pipeline

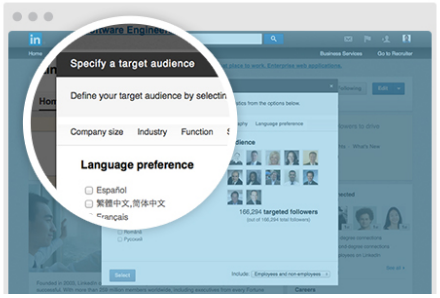


Figure 5: Supported languages at LinkedIn. The site is available in 20 languages, including Chinese, English, French, German, Italian, Portuguese, etc.

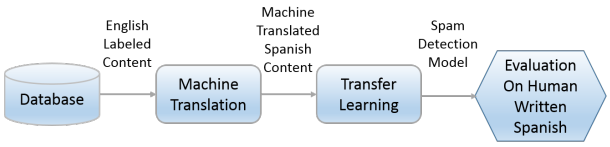


Figure 6: Proposed pipeline for Spanish spam detection. Machine Translator is employed to translate the English labeled content into Spanish. We adopt transfer learning to build a spam detection model. Evaluation is done on human written Spanish.

contains four steps, as shown in Figure 6. To make use of labeled English samples in the server, we collect the English labeled data for translation in the first step. During the collection step, we first apply the language detector [12] to make sure that the selected samples are written in English. After collecting a large set of labeled English content, we use machine translation tools to translate the English content into Spanish. We select the Google translate API in the second step as it is one of the most popular machine translators. The second step results in a large set of machine translated Spanish content, which can be considered as the training samples for building a Spanish spam detector. Due to the distribution difference between the machine translated Spanish and the human written Spanish, we propose a two-step transfer learning algorithm to generate a Spanish spam classifier. Finally, we evaluate our model on hunman written Spanish content.

3.1 Feature Generation

In the aforementioned pipeline, a collection of machine translated Spanish data is available after the machine translation. We generate tf-idf features to represent both machine translated Spanish and human written Spanish data. The features are then used for building transfer learning models.

Term frequency-inverse document frequency (tf-idf) is a numerical statistic that is intended to reflect how important a word is to a document in a collection of corpus. The tf-idf value increases proportionally to the number of times a word

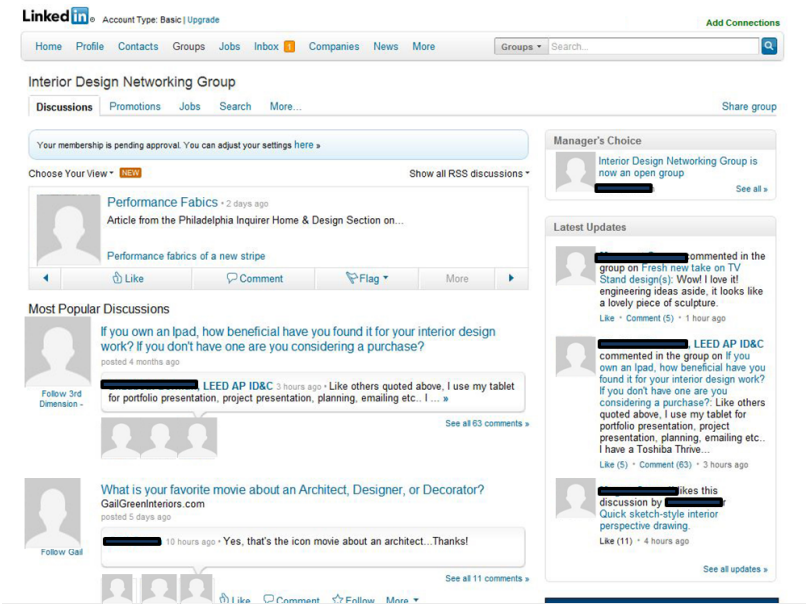


Figure 4: One example of group discussions at LinkedIn. Members in this group interact with each other by posting questions about interior design and answering other members' question.

appears in the document, but is offset by the frequency of the word in the corpus. The offset is necessary since some words are generally more common than others.

Tf-idf is the product of two statistics, including term frequency (tf) and inverse document frequency (idf). Various ways for determining the exact values of both statistics exist. We apply the Scikit-learn (`http://scikit-learn.org/stable/`) [16] package to generate the tf-idf features, where tf, idf and tf-idf are defined as follows:

- $tf(t,d) = 1$ if term t occurs in document d.
- $idf(t,D) = log\frac{N}{|d \in D: t \in d|}$ where N is total number of documents in the corpus, and $|d \in D : t \in d|$ is the number of documents that include term t.
- $tfidf(t,d,D) = tf(t,d) \times idf(t,D)$.

A high weight in tf-idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms.

3.2 Feature selection

The vocabulary of the corpus is large, and the data matrix for the tf-idf features is sparse. To study the distribution of machine translated Spanish and human written Spanish, we first select the top features using the labels.

Here, we apply Lasso [22] to do the feature selection. Lasso is essentially a linear regression problem with an l_1 norm regularizer. It is often adopted as an efficient tool to perform the feature selection and its extension remains as a hot topic in recent research [27, 25]. Denote the feature matrix as A, the label as y, and the model as x. Lasso can be formulated as the following optimization problem:

$$\min_x \frac{1}{2}\|Ax - y\|_2^2 + \lambda\|x\|_1 \quad (1)$$

where λ is a parameter that controls the balance between the loss function and the penalty. The Lasso problem can be solved efficiently by FISTA algorithm [3]. After we get the solution x, the non-zero entries in x correspond to the selected features in A.

We apply Lasso to both machine translated Spanish and human written Spanish to investigate the characteristics of these two domains. The top 20 selected features are listed in Table 1. It can be verified that most of the features relate to jobs. We can see there are some top features shared by machine translated Spanish documents and human written Spanish documents, but about half of the features are different between the two domains. This is not surprising: the machine translated jobs and the manually constructed jobs were taken from non-overlapping data sources. The former was translated job posts from English jobs on LinkedIn, while the latter was obtained by crawling the web and finding job posts written in Spanish. While the top 20 features in both data sets are related to the broad topic of jobs, e.g. business, information, employee, opportunity, development, team, management (Table 1), we cannot expect the features to be 100% identical over both data sets. We only expect stop words to be invariant across different data sets, not nouns, verbs, or adjectives. Furthermore, some of the top 20 features in the manually constructed and the machine translated sets are morphological cognates such as **buscamos** = **we look** and **buscando** = **looking**; **cliente** = **client** and **clientes** = **clients**, which further illustrates that the top 20 features are semantically very close in both data sets.

Despite the above issues, the machine translated content has the same overall intended meaning as the human written content. In addition, as we select more unigram features, we find larger overlaps of selected features between the two domains. Thus, transfer learning is a viable approach to solve the problem. Consider the machine translated Spanish as the source domain, and the human written Spanish as the target domain, transfer learning aims to borrow the

Table 1: unigram features for two domains

Machine Translated Spanish		Human written Spanish	
Selected Spanish Features	English translation	Selected Spanish Features	English translation
años	years	buscamos	look for
buscando	search for	clientes	customers
cliente	customer	comercial	business
clientes	customers	conocimientos	background
desarrollo	development	cv	cv
empresa	company	desarrollo	development
empresas	companies	gestión	management
equipo	team	equipo	team
están	are	españa	Spain
experiencia	experience	experiencia	experience
favor	please	funciones	functions
gestión	management	manejo	management
http	http	millones	million
información	information	más	more
mundo	world	proyectos	projects
más	other	responsable	responsible
oportunidad	opportunity	seguimiento	follow up
posición	position	turismo	travel
servicios	services	ventas	sales
web	web	área	area

knowledge learnt from source domain and apply it to target domain. The details of our two-step transfer learning algorithm are presented in the next section.

4. TRANSFER LEARNING

In this paper, we propose a two-step transfer learning algorithm to learn the content classifier from machine translated Spanish and apply it to human written Spanish. To our best knowledge, this is the first work that transfers the knowledge from machine translation corpus to human written corpus in social network. This paper is different from a related two-stage transfer learning algorithm [20] in the sense that the number of domains are different. The previous work [20] focuses on multiple source domain transfer learning, in which the second stage utilized the smoothness assumption among hypotheses from multiple sources. However, this paper focuses on single source domain transfer learning where the smoothness assumption cannot be established. The proposed algorithm is a general algorithm that can be adopted to any other languages as well as any binary classification problem.

Denote the machine translated Spanish as the source domain T, and the human written Spanish as the target domain H. We have shown in Section 3 that the distributions of the T and H could be different while the support can be the same with the increase of feature dimension. Therefore, we make an assumption that the distribution difference mainly lies in the marginal probability rather than the conditional probability. Based on the assumption, we propose our first step of transfer learning: instance re-weighting. The main idea of instance re-weighting is to calculate the weights for the data in the source domain, and re-weight the source domain for matching the marginal distribution with the target domain.

After the first step, we train a logistic regression model on the re-weighted source data, then apply it to the target domain. In addition, if there are some labels available in the target domain, we propose an adaptive learning algorithm as the second step to update our model learnt in the first step. The adaptive learning algorithm penalizes the misclassification errors in the target domain while minimizes the modifications of the model. The second step will also make up the conditional distribution difference between T and H if our assumption in the first step does not hold.

4.1 Instance Re-weighting

The first step in the proposed transfer learning framework is the Kernel Mean Matching (KMM) algorithm [9], which is a nonparametric method to directly infer weights for source domain without distribution estimation. KMM utilizes the unlabeled data to build a bridge for connecting two domains. In general, the estimation problem with two different distributions $Pr(x, y)$ and $Pr'(x, y)$ is unsolvable, as the two terms could be arbitrarily far apart. In particular, for arbitrary $Pr(y|x)$ and $Pr'(y|x)$, there is no way to infer a good estimator based on the available training samples in the source domain. Hence a simplified version makes assumption that $Pr(x, y)$ and $Pr'(x, y)$ only differ via $Pr(x)$ and $Pr'(x)$, i.e., the conditional probability remains unchanged: $Pr(y|x) = Pr'(y|x)$. This particular case of transfer learning has been termed as covariate shift.

The basic assumption behind KMM approach is that between the two domains, the key difference lies in the marginal distribution $Pr(x)$ rather than the conditional distribution $Pr(y|x)$. The idea of KMM is to find the appropriate weights for T which minimize the discrepancy between the mean values of T and H in the common projected space. Denote x_i as the i-th sample in T and x'_i as the i-th sample in H. The mean differences after projection can be formulated as:

$$\min_{\alpha} \left\| \frac{1}{m} \sum_{i=1}^{m} \alpha_i \Phi(x_i) - \frac{1}{m'}, \sum_{i=1}^{m'} \Phi(x'_i) \right\|^2 . \quad (2)$$

where α_i $(i = 1, \ldots, m)$ is the weight we want to learn, m is the number of samples in T, m' is the number of samples in H, and Φ is a mapping which maps the raw features into a latent space.

The formulation is actually a constrained QP (Quadratic Programming) problem. After weighting, we still want to ensure $\alpha Pr(x)$ is close to a distribution, so there are two constraints that need to be added: $\alpha_i \in [LB, UB]$ and $|\frac{1}{m}\sum_{i=1}^{m}\alpha_i - 1| \leq \epsilon$, where LB and UB is the lower bound and the upper bound of α_i respectively. Denote $K_{ij} = k(x_i, x_j)$ and $\kappa_i = \frac{m}{m'}\sum_{j=1}^{m'} k(x_i, x'_j)$. We can rewrite the formulation (2) as:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{m^2}\alpha^T K\alpha - \frac{2}{m^2}\kappa\alpha, \\ \text{s.t.} \quad & \left|\frac{1}{m}\sum_{i=1}^{m}\alpha_i - 1\right| \leq \epsilon, \\ & LB \leq \alpha_i \leq UB. \end{aligned} \tag{3}$$

It can be solved by many optimization algorithms, such as interior point methods or any other successive optimization procedure. We simply apply the **optimize** module in **SciPy** to calculate the solution.

After we obtaining α, we multiply each sample in T with the corresponding α_i, then build a regularized logistic regression model w_0 as follows:

$$\min_{w_0}\sum_{i=1}^{m} log(1 + \exp(-y_i w_0^T \alpha_i x_i)) + \lambda\|w_0\|^2. \tag{4}$$

If there is no labeled data available in target domain, we apply the model learnt in Equation (4) to the target domain for Spam detection. There are many different classification models can be adopted to solve this problem, such as Naive Bayes [11] and SVM [28]. The reason of choosing the logistic regression to build our classification model lies in three folds: (1) Logistic regression has achieved great success in many real-world applications; (2) The processing time for an online product should be as small as possible, and logistic regression is a perfect match; (3) It is easy to be generalized to a distributed version.

4.2 Adaptive Learning

KMM targets at the scenario that there is no labeled data available in the target domain. So far, we have not touched any label information in the target domain. There are scenarios under which a small portion of labeled target data $\{x''_i, y''_i | i = 1, \ldots, m''\}$ is available. For example, the samples may be labeled by crowdsourcing. Next, we show how to improve our model w_0 learnt from KMM by using additional labeled instances in H. We propose an adaptive learning strategy, which aims to learn a perturbation $\Delta f(x) = \Delta w^T \phi(x)$ on the basis of w_0 to further compensate for domain mismatch. Denote the classifier we learnt through KMM as w_0, the target classifier we aim to learn as w, we have the following relation: $w = w_0 + \Delta w$.

Based on the aforementioned assumption that the difference between the distributions of machine translated Spanish and human written Spanish mainly lies in the marginal distribution, the intuition of the proposed adaptive learning is that the difference of w and w_0 should be as small as possible. There are related previous work [6, 8], which were proposed to learn a perturbation of the source hyperplane,

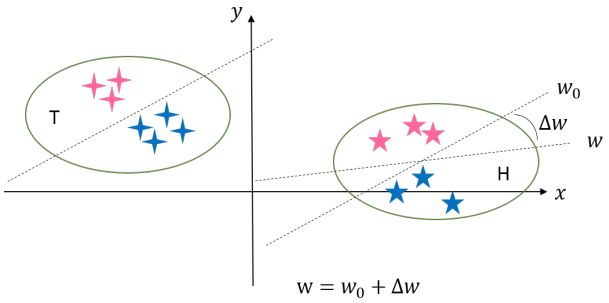


Figure 7: Illustration of adaptive learning. Model w_0 is borrowed from the domain T, and the model w for the domain H should be similar to w_0. Δw is a small perturbation we aim to learn.

by minimizing the classification error on labeled examples in the target domain. The pertubation can also be considered as new feature representations that correct for the domain change. The intuition of our adaptive learning algorithm is consistent with the online learning, thus this step can be put into online learning system.

Our adaptive learning step aims to learn a perturbation Δw such that the difference between prediction $f(x) = w^T x$ and $f_0(x) = w_0^T x$ is small (Figure 7). The following equation holds: $f(x) = f_0(x) + \Delta f(x) = w_0^T \phi(x) + \Delta w^T \phi(x)$, where Δw is the difference to be learnt here. Therefore, we formulate the adaptive learning as the following optimization problem:

$$\begin{aligned} \min_{\Delta w} \quad & \sum_{i=1}^{m''} \log(1 + exp(-y''_i w^T x''_i)) + \lambda\|\Delta w\|_2^2 \\ =\min_{\Delta w} \quad & \sum_{i=1}^{m''} \log(1 + exp(-y''_i (w_0 + \Delta w)^T x''_i)) + \lambda\|\Delta w\|_2^2, \end{aligned} \tag{5}$$

where the logistic loss is utilized with the penalty that encourages the perturbation $\|\Delta w\|$ to be small. Here λ is a tuning parameter to control the balance between the loss function and the penalty. Problem (5) involves a convex smooth function, which can be solved efficiently by gradient-based method. Adaptive learning can make up for the potential issue that there are also conditional differences in the distributions between the machine translated Spanish and human written Spanish. It can also serve as an online update algorithm when more and more labeled target data become available.

Finally, we obtain the model $w = w_0 + \Delta w$ after the proposed two-step transfer learning algorithm. We summarize our two-step transfer learning algorithm in Algorithm 1.

5. EXPERIMENTS

In this section, we demonstrate the effectiveness of our proposed machine learning pipeline using production data at LinkedIn.

Following the pipeline 6, we collect 12000 pieces of English content in group discussions at LinkedIn. The content is collected from the groups where job posts are not allowed, i.e., job related content are considered as spam in these groups. The English content is kept balanced for the two classes: job-related posts and non-job posts. It is used for translation

Algorithm 1 A two-step transfer learning algorithm for bilingual spam detection

Require: a large set of labeled samples in T: $\{x_i, y_i\}$, a large set of unlabeled samples in H: $\{x'_i, y'_i\}$, (optional) a small set of labeled samples in H: $\{x''_i, y''_i\}$, $\Delta w = 0$

Ensure: w

1: Calculate weights α by solving

$$\min_{\alpha} \left\| \frac{1}{m} \sum_{i=1}^{m} \alpha_i \Phi(x_i) - \frac{1}{m'} \sum_{i=1}^{m'} \Phi(x'_i) \right\|^2$$

2: Compute w_0 by solving

$$\min_{w_0} \sum_{i=1}^{m} log(1 + \exp(-y_i w_0^T \alpha_i x_i)) + \lambda \|w_0\|^2$$

3: (Optional) Compute Δw by solving

$$\min_{\Delta w} \sum_{i=1}^{m''} \log(1 + exp(-y''_i (w_0 + \Delta w)^T x''_i)) + \lambda \|\Delta w\|_2^2$$

4: **return** $w = w_0 + \Delta w$

into Spanish to form the source domain. We translated all the collected human labeled content from English to Spanish with Google translate API.

In addition, we also collect 12000 pieces of human written Spanish content (balanced) by crowdsourcing, which serves as the target domain. Inside the target domain, there are 50 pieces of comments considered as labeled samples used in our adaptive learning step and the remaining ones are treated as test data for evaluation. We generate tf-idf features for both machine translated Spanish and human written Spanish, and it turns out that the feature spaces of the two domains differ from each other, as shown in Table 1.

5.1 Experimental Settings

Based on the source domain corpus and the target domain corpus, we first generate the tf-idf features for each domain, and then use the intersection of selected features to build the Spanish spam detector.

Scikit-learn (`http://scikit-learn.org/stable/`) [16] is a powerful machine learning package in python which provides feature generation, feature selection as well as classification functions. We make use of scikit-learn to carry out the experiment: first, we generate the tf-idf features for both domains by use of *feature_extraction.text.TfidfTransformer* class; then we build the classification model via *linear_model.LogisticRegression* class.

In our experiment, it turns out that stemmer would not help in classification tasks, as it reduced the feature dimension of the problem. There are overlapped unigram features between machine translated Spanish and human written Spanish, but the feature spaces of these two domains are not identical. We simply choose the intersection as our feature space to conduct the transfer learning.

We implement our proposed two-step transfer learning algorithm in Python, and we solve the optimization problem in both KMM and adaptive learning by calling the scipy.optimize module (`http://docs.scipy.org/doc/scipy/reference/optimize.html`). We increase the tf-idf feature dimension to leverage the loss of information introduced by performing intersection. To verify the efficacy of the proposed two-step transfer learning algorithm, we compare it with four other algorithms:

- Baseline: We train a model on T, and directly apply it to the test data in H.
- SLT (Small Labeled dataset for Training): We train a model on a small portion of labeled data in H (50 samples), and apply it to the test data in H.
- KMM: We calculate the weights for data in T via KMM, then train a model on weighted data in T, and apply it to the test data in H.
- Adaptive Learning: We apply the adaptive learning (Equation (5)) to build a model on the basis of model generated from Baseline, and apply it to the test data in H.

We choose the above four algorithms for comparison because they are both straight forward and practically viable algorithms for an online product. To further test the robustness of different approaches, we conduct our experiments by varying the dimension of feature space as follows:

- We generate 600 features for both T and H, and use the intersection (380 features) as the feature space.
- We generate 800 features for both T and H, and use the intersection (504 features) as the feature space.
- We generate 1000 features for both T and H, and use the intersection (629 features) as the feature space.
- We generate 2000 features for both T and H, and use the intersection (1260 features) as the feature space.

From our experience, the more features we generated, the better it will generalize to unseen data.

5.2 Experimental Results

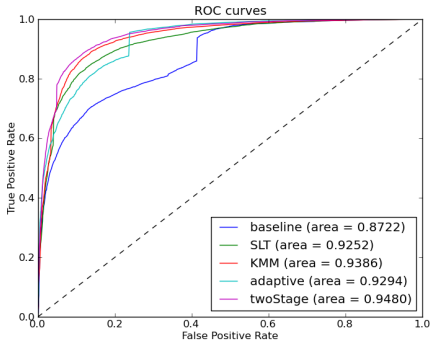
The results of all five algorithms are shown in 8 and Table 2. To encourage the model to achieve the desired performance for production, we set the threshold of logistic regression so as to make the precision fixed at 0.95.

The experimental results exhibit the efficacy of our proposed algorithm in the following aspects:

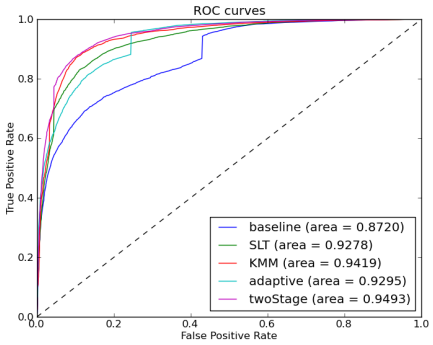
- SLT works better than Baseline. The reason is that baseline trains a model in the source domain and tests it in a different domain, while SLT trains and tests the data from the same domain. The domain difference makes the baseline perform worse even with a large amount of training data.
- KMM works better than SLT. The reason lies in two folds: KMM trains the model based on the re-weighted source domain data, which leverages the difference between the two domains. In addition, the training data size of KMM is much larger than that of SLT.
- Adaptive learning works better than SLT, but worse than KMM. Since the reference model in adaptive learning here comes from baseline, which ignores the distribution difference between the two domains. However, it utilizes more information from the related source domain, which makes it perform better than SLT.

Table 2: Results for different dimensions of features

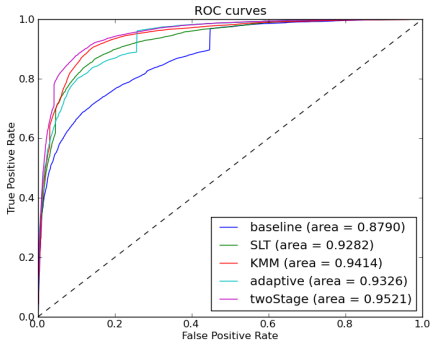
tf-idf dimension	intersection dimension	algorithm	precision	recall	f1-score	AUC
600	320	baseline	0.95	0.44	0.60	0.87
		SLT	0.95	0.42	0.63	0.93
		KMM	0.95	0.46	0.64	0.94
		adaptive	0.95	0.57	0.71	0.93
		twoStage	0.95	0.65	0.77	0.95
800	504	baseline	0.95	0.47	0.63	0.87
		SLT	0.95	0.51	0.66	0.93
		KMM	0.95	0.66	0.78	0.94
		adaptive	0.95	0.57	0.71	0.93
		twoStage	0.95	0.65	0.77	0.95
1000	629	baseline	0.95	0.46	0.62	0.88
		SLT	0.95	0.52	0.67	0.93
		KMM	0.95	0.66	0.78	0.94
		adaptive	0.95	0.61	0.74	0.93
		twoStage	0.95	0.79	0.86	0.95
2000	1260	baseline	0.95	0.50	0.65	0.89
		SLT	0.95	0.58	0.72	0.93
		KMM	0.95	0.73	0.82	0.95
		adaptive	0.95	0.70	0.80	0.94
		twoStage	0.95	0.82	0.88	0.96



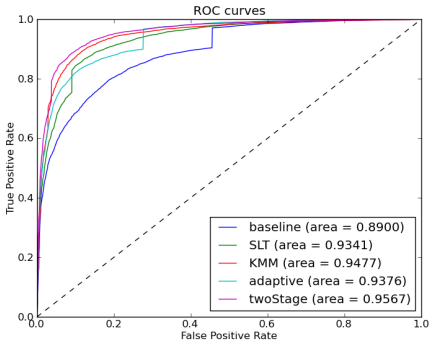
(a) AUC curves for 600 features



(b) AUC curves for 800 features



(c) AUC curves for 1000 features



(d) AUC curves for 2000 features

Figure 8: Experiment results for varied dimensions of features: 600, 800, 1000 and 2000. In all these settings, our proposed algorithm achieves the best performance compared with other algorithms.

- Our proposed two-step algorithm produces the best performance, as it combines the advantages of KMM and adaptive learning.

From the results, we conclude that each of the two steps (Instant Weighting and Adaptive Learning) improves the performance compared to the baseline method, while combining the two steps works the best. As the feature di-

mension increases, the performance also improves. To avoid overfitting and ensure the efficiency as an online product, we stop at 2000 features.

The results also verify our assumption about the distributions of two domains, that is, the main difference lies in the marginal distribution rather than the conditional difference. The proposed two-step transfer learning algorithm achieves the required performance for production.

5.3 Use of less unlabeled samples in H

In real applications, we may not have enough labeled samples in H. In addition, we may not have enough unlabeled samples in H in advance for KMM. As the mean value of projected distribution for the target domain cannot be captured, KMM may fail to match the difference between the source domain and target domain.

To verify whether the KMM algorithm works with a small sample size in the target domain, we further design a series of experiments with varied sizes of samples in the target domain. Specifically, We vary the amounts of unlabeled samples from 20 to 1000 in human written Spanish to test the KMM algorithm, and the results can be seen in Table 3.

Table 3: KMM results for a small number of samples in the target domain

sample size in H	precision	recall	f1-score	AUC
20	0.95	0.28	0.44	0.89
50	0.95	0.56	0.71	0.93
100	0.95	0.62	0.75	0.94
200	0.95	0.68	0.79	0.95
500	0.95	0.82	0.82	0.96
1000	0.95	0.86	0.86	0.97

We want to emphasize that even with a small sample size (100) in H, we can get the desired performance for production. The explanation behind the experiment results lies in the fact that as long as the target samples are balanced, KMM can capture the mean value of the distribution in the projected space even with small sample size. This is critical for an online product: as in the real world, we want to ensure the model to work even with small amounts of human written content. As more and more human written Spanish data become available, we can apply the second step to incrementally update the model, which can also be considered as online learning.

6. RELATED WORK

Cross-lingual content classification is an important topic in natural language processing area, and it gains more attention as the social network becomes global [1, 17, 14, 13, 15, 18, 24, 4]. The transfer learning research on text mining [5, 10] mainly focus on transferring the knowledge from different topics in the same language. The work done by Vinokourov [23] can be considered as the unsupervised cross-lingual learning, where the correlation between two languages are learnt by Canonical Correlation Analysis. In this work, the high correlation between English and French words indicates that they represent the same semantic information. It is different from our work as we deal with supervised learning problems. There is related research that focus on modeling the multi-view representations in which each language is considered as a separate source, and a joint loss is minimized while the consistency between languages is ensured [1]. One related work [15] also represents the data from two languages views and proposes a non-negative matrix tri-factorization (BNMTF) model for the cross-lingual sentiment classification problem. These methods aim to utilize the labeled data in two languages and build a joint model for multi-view learning, which differs from our work as we do not have labeled data in both languages, and for an online product, it is not practical to translate every piece of message into the other language.

The most relevant work was done by Ling et al. [13]. In their work, they translated the Chinese web pages into English, and the common parts of two languages are extracted and used for classification. Specifically, the KL divergence between the feature distribution and the label distribution is minimized to generate the classification model. This work bridges the translated English and label set by common features, and uses a parametric algorithm to build the classifier. In our work, we make use of non-parametric transfer learning algorithm to migrate the information learnt from the machine translated Spanish and apply it to the human written Spanish. To the best of our knowledge, this is the first attempt to take care of the distribution difference between the machine translated corpus and human written corpus in social network. Moreover, we propose an adaptive learning algorithm to update the model online.

7. CONCLUSIONS

In this paper, we develop a machine learning pipeline to tackle the multilingual spam detection problem at LinkeIn. We propose to build a model pool in which one model works for one language, and we start with Spanish as it is widely used in US. Due to the lack of labeled data in Spanish, we first translate the English labeled content into Spanish content via machine translation. Since the distributions of machine translated Spanish and human written Spanish are not identical, we propose a two-step transfer learning algorithm to transfer the knowledge learnt from machine translated Spanish to human written Spanish. In the first step, KMM is applied in order to match the marginal distribution difference between the two domains. If there is a small portion of labeled data available in the human written Spanish, we propose a second step to incrementally update the model obtained from the first step. Results show that our proposed pipeline achieves the requirements in Spanish spam detection for industrial products.

The proposed pipeline is a generic system which can be applied to any other languages by substituting the target lanaguage in machine translation. We plan to extend our proposed algorithms to solve the French and Chinese spam detection problems in the future. In addition, the classification pipeline can deal with different kinds of classification problems as long as there are labeled samples available. We also plan to apply the pipeline to other text classification problems at LinkedIn. Furthermore, the adaptive learning algorithm we propose in the second step can serve as an online learning model. When more labeled data in Spanish become available, the model will be adaptively updated.

8. REFERENCES

[1] M.-R. Amini and C. Goutte. A co-classification approach to learning from multilingual corpora. *Machine learning*, 79(1-2):105–121, 2010.

[2] C. Banea, R. Mihalcea, and J. Wiebe. Multilingual subjectivity: are more languages better? In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 28–36. Association for Computational Linguistics, 2010.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] W. De Smet, J. Tang, and M.-F. Moens. Knowledge transfer across multilingual corpora via latent topics. In *Advances in Knowledge Discovery and Data Mining*, pages 549–560. Springer, 2011.

[5] C. Do and A. Y. Ng. Transfer learning for text classification. In *NIPS*, 2005.

[6] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank. Domain transfer svm for video concept detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1375–1381. IEEE, 2009.

[7] G. Foster, C. Goutte, and R. Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459. Association for Computational Linguistics, 2010.

[8] J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell. Efficient learning of domain-invariant image representations. In *International Conference on Learning Representations*, 2013.

[9] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2006.

[10] J. Jiang. *Domain adaptation in natural language processing*. ProQuest, 2008.

[11] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes. Multinomial naive bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence*, pages 488–499. Springer, 2005.

[12] D. Kurokawa, C. Goutte, and P. Isabelle. Automatic detection of translated text and its impact on machine translation. *Proceedings. MT Summit XII, The Twelfth Machine Translation Summit International Association for Machine Translation Hosted by the Association for Machine Translation in the Americas*, 2009.

[13] X. Ling, G.-R. Xue, W. Dai, Y. Jiang, Q. Yang, and Y. Yu. Can chinese web pages be classified with english data source? In *Proceedings of the 17th international conference on World Wide Web*, pages 969–978. ACM, 2008.

[14] X. Meng, F. Wei, X. Liu, M. Zhou, G. Xu, and H. Wang. Cross-lingual mixture model for sentiment classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 572–581. Association for Computational Linguistics, 2012.

[15] J. Pan, G.-R. Xue, Y. Yu, and Y. Wang. Cross-lingual sentiment classification via bi-view non-negative matrix tri-factorization. In *Advances in Knowledge Discovery and Data Mining*, pages 289–300. Springer, 2011.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] J. C. Platt, K. Toutanova, and W.-t. Yih. Translingual document representations from discriminative projections. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 251–261. Association for Computational Linguistics, 2010.

[18] P. Prettenhofer and B. Stein. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127. Association for Computational Linguistics, 2010.

[19] J. Slocum. A survey of machine translation: its history, current status, and future prospects. *Computational Linguistics*, 11(1):1–17, 1985.

[20] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye. A two-stage weighting framework for multi-source domain adaptation. In *Advances in neural information processing systems*, pages 505–513, 2011.

[21] J. Tang, X. Wang, H. Gao, X. Hu, and H. Liu. Enriching short text representation in microblog for clustering. *Frontiers of Computer Science*, 6(1):88–101, 2012.

[22] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[23] A. Vinokourov, N. Cristianini, and J. S. Shawe-taylor. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in neural information processing systems*, pages 1473–1480, 2002.

[24] C. Wan, R. Pan, and J. Li. Bi-weighting domain adaptation for cross-language text classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1535, 2011.

[25] J. Wang, J. Zhou, P. Wonka, and J. Ye. Lasso screening rules via dual polytope projection. In *Advances in Neural Information Processing Systems*, pages 1070–1078, 2013.

[26] P. Wang and C. Domeniconi. Towards a universal text classifier: Transfer learning using encyclopedic knowledge. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 435–440. IEEE, 2009.

[27] S. Xiang, T. Yang, and J. Ye. Simultaneous feature and feature group selection through hard thresholding. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 532–541. ACM, 2014.

[28] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th international conference on Multimedia*, pages 188–197. ACM, 2007.