A Language-Modeling Approach to Health Data Interoperability

Matthew Michelson, Steven N. Minton and Kane See
InferLink Corporation
2361 Rosecrans Avenue, Suite 348
El Segundo, California 90245

Abstract

The need for health providers to share information is a pressing need in our ever more connected world. A patient's health information should seamlessly flow from labs to hospitals to primary care offices. To address this need, in this paper we present the Health E-Match, which focuses on the matching health terms in support of semantic interoperability. Health E-Match determines the semantic similarity between data items, realizing, for instance, that "BHGC (UR)" and "BETA-HCG (QUAL)" both refer to the same pregnancy test, known as "Beta human chorionic gonadotropin, urine qualitative." Our approach is grounded in probabilistic machine learning, and leverages several sophisticated methods for comparing the similarity between medical data items beyond simple edit distance. We present two large scale, real-world experiments to verify that our approach is both accurate and has the ability to eventually be "universal" in that models trained on one set of data translate to strong performance on data from a completely different provider.

Introduction

There is a pressing need for semantic "interoperability" between health systems. Imagine a scenario where a patient in the emergency department has a test the hospital calls a "Beta human chorionic gonadotropin, urine qualitative"[1] test. Yet, when she follows up with her family doctor, the local record system at the doctor's office calls the test "BHCG (UR)," and so it cannot receive the result from the ED. So, the doctor orders the test at a lab, who names the test locally as "HCG QUALITATIVE." (We emphasize, these are *real-world* names for this test from real hospital systems). The various names for the same test therefore pose challenges for integrating and communicating this result. The need to communicate and integrate data across providers is particularly vexing in the medical domain, since having access to good information is critical for diagnosing illness and developing and maintaining appropriate treatment plans. Unfortunately, current progress on this problem is painstakingly slow, even with government urging to address the problems.

[1]This is a pregnancy test.

Much of the current (government supported) solution to this problem focuses on providers adopting standards, mapping their local data terminologies to standard ontologies. However, beyond the obvious issue involving cost (and potential error) when requiring people to do their own terminology mapping to the standard, there are a number of more subtle issues with this approach.

First, a single standard is often not well suited to represent the breadth and depth of medicine and biology. Sometimes there are multiple standards, each appropriate in different situations. For instance, for radiology, some users prefer the RadLex Playbook, provided by the Radiological Society of North America, while for laboratory tests, users prefer LOINC. But there are potential difficulties in integrating these sources into a master ontology given their representations (for instance, RadLex is more of a dictionary, rather than a full ontology of concepts). Overall, the development and use of standards presents a number of complexities that are still a long way off from being solved.

Second, while we hope (and believe) that there will be an increasing amount of standardized data that is exchanged, it is important to realize that for the foreseeable future there will be many data sources that haven't (yet) been standardized, including sources of archived data. We call this "local-to-local" interoperability (versus "local-to-standard") and it comes about in many practical situations. For instance, consider integrating data for public health from a number of rural hospitals, each of which is small, uses their own terminology, and have limited IT resources for interoperability. Rather than each mapping to a standard for integration, it might require each hospital mapping to each other, since there might already be partial mappings in place historically, and redoing a full mapping is too costly.

Third, mapping the data to a standard value is often a non-trivial operation. Consider Table 1 which shows 14 real-world representations for our example test "Beta human chorionic gonadotropin, urine qualitative." These variations involve a number of linguistic transformations to map the values including various acronyms (BHCG, Beta-HCG, hCG), abbreviations (UR and URN for urine), and even synonyms ("Pregnancy test" and "Urine Pregnancy"). This table demonstrates the level of sophistication required to do these mappings, even for a single test.

Finally, even when dealing with the myriad of difficulties

Table 1: Real-world representations of the same test

Beta human chorionic gonadotropin (beta-HCG), urine qualitative
Bedside Pregnancy Test/OP ONLY
Beta hCG Qualitative Urine
BHCG (UR)
BHCG Qual
HCG QUAL, URN
HCG QUALITATIVE
HCG Qualitative Urine
hCG Urine
pregnancy qual urine beta hcg
Urine Beta hCG Qualitative
URINE HCG PREGNANCY
Urine Pregnancy

when mapping the data, there is an added level of difficulty since in many cases the mapping involves multiple concepts simultaneously. For instance, the data item"XR Toes Great Left" involves both a procedure, "X-Ray," and a body part, the left great toe. To map this order into a standard requires the ability to support multiple, simultaneous concept mappings.

In this paper we describe an approach, under development, called Health E-Match that supports the exchange of health data by learning how to map medical terms that refer to the same procedure, medicine, order, etc. Further, using large volumes of real-world data, we demonstrate the potential effectiveness of our approach. The long-term goal is to enable automatic mappings between ontologies, terminologies, and other knowledge representations that when done would allow for true exchange of health information across systems. We believe this capability is essential if universal healthcare information exchange is to be achieved, since local "non-standard" terminologies are used by so many healthcare organizations (and so much historical data exists using these local terminologies).

The rest of this paper is organized as follows. First we describe our algorithm and the overall system architecture of Health E-Match. Then, we describe our experiments and the results, followed by a discussion of related efforts in mapping healthcare terminologies. Next we provide some discussion, and we conclude with our final thoughts.

The Health E-Match System

The crux of our technical approach uses probabilistic reasoning to reason about the similarities and differences between terms, coupled with statistical machine learning methods to develop the appropriate mapping across a given language pair. Our approach is motivated by natural language translation techniques, such as those for translating Russian to English, French to Arabic, etc. (Koehn 2010). While probabilistic approaches may appear to introduce uncertainty, in fact, probabilistic reasoning often outperforms traditional, rule-based approaches, which tend to be rigid (Sebastiani 2002). Manually developed rule-based systems tend to be

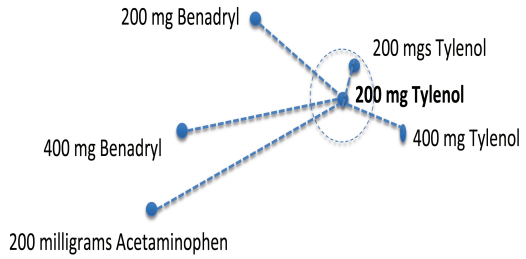

Figure 1: Edit distance can be flawed

difficult to develop and brittle to maintain. As the complexity of a rule base grows, interactions between rules become unpredictable, leading to failures and maintenance issues. In contrast, probabilistic inference systems often handle "corner cases" more gracefully, and perhaps most importantly, they work well in conjunction with statistical machine learning algorithms, which can often produce highly accurate probabilistic rules based on large quantities of data.

One of the key contributions of our learning system is in leveraging various types of terminology mappings to determine the best translation. Some of the mappings are based on transitional word transformations such as acronyms (e.g., "beta-HCG" and "BHCG" for "Beta human chorionic gonadotropin.") Other mappings are based upon word-substitution transformations such as the substitutions of "urine," "ur," and "urn" in Table 1. Transformations for mapping can also represent more sophisticated relationships between terms, such as describing ontological relationships. For example, "Celiprolol" is a type of "beta-blocker," and "Cardem" is a brand name for "Celiprolol." Therefore, in certain cases these may be substituted for one another. Yet another transformation involves similar concepts, such as "blood co-oximetry," and "arterial blood gas," which are both blood tests measuring certain levels in the blood. In this case, these might warrant matching or not (depending on the context).

Mapping health terms can be described as a function that probabilistically relates textual items written in a source language S to items in a target language T. For our purposes, we will consider a wide variety of health languages, including vocabularies, order sets, ontologies etc., and thus the items that we map may be terms, phrases, concepts, etc.[2]

One of the key issues when determining whether an item s in source language S maps to an item t in target language T is how to evaluate the similarities and differences between s and t. One of the classic approaches is to use metrics such as edit distance as a proxy for similarity. Basic edit distance approaches simply count the number of edits (e.g., character differences) required to change the item s into t. However,

[2]We note that we are not focusing on a specific data model or representation (e.g., RDF vs. XML). While these can be important for developing an appropriate and expressive ontology, our goal is to develop a translator that can map languages regardless of their specific representational scheme.

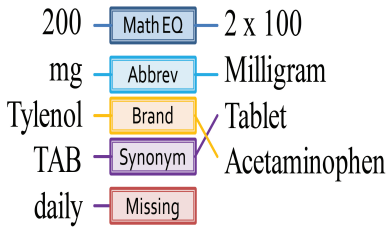

Figure 2: Example transformations required to map between terms in two languages

this naive approach does not work well for many applications. As shown in Figure 1, in terms of edit distance, "200 mg Tylenol" is much closer to "200 mg Benadryl" than it is to "200 milligrams Acetaminophen," even though the latter is a better match.

In contrast, our approach evaluates similarity based on the *transformations* mapping one data element to another (Minton et al. 2005). Figure 2 shows the relationship between the source term"200 mg Tylenol TAB daily" and the target term "2x100 Milligram Tablet Acetaminophen," in terms of a set of transformations, including a brand transformation that maps "Tylenol" to "Acetaminophen," a synonym transformation that converts "TAB" to "Tablet," an abbreviation transformation that relates "mg" and "Milligram," and so on. These transformations can be functionally defined (e.g, such as equality or misspellings) or based upon knowledge-rich representations such as databases or ontologies. In the medical domain, we can ingest databases of synonyms ("thoracic" is "chest"), generic drug names and their brand names ("Lisinopril" is "Zestril"), common abbreviations ("millligram" is "mg"), as well as common ontologies such as SNOMED.

Given this framework for assessing similarity, we then use a generative language modeling approach to derive probabilities where, for every source element, we consider the possible target elements that it could map to (e.g., we assess the probability that any t_i in T is the "right" mapping for our input term s). Formally, for any given source element s and target element t, let $P(t|s)$ be the probability that s maps to t. Using Bayes rule, we can write this as follows:

$$P(t|s) = P(t)P(s|t)/P(s)$$

Since our interest is in ranking the target elements in order to map a given source element and the denominator is constant for any given source element, we can follow the standard approach and drop the denominator from our calculations, giving:

$$P(t|s) \propto P(t) * P(s|t)$$

The first term, $P(t)$ is referred as the "prior," and reflects the a priori probability that the target element t should be mapped, independent of the source element s. For instance, in a target vocabulary, there are often terms that are more commonly used, and these would have a higher prior. These priors can be directly estimated from training data.

The second term, $P(s|t)$ or the "likelihood" term, is the conditional probability of s given t. Essentially, we can think of the target element as "generating" possible source elements. Let us denote the set of transformations that convert t to s as $\{\Delta_1, \Delta_2, \ldots, \Delta_k\}$. We can model $P(s|t)$ by $P(\Delta_1, \Delta_2, \ldots, \Delta_j)$. In other words, the probability of generating s from t is given by the probability of the corresponding set of transformations. This can be approximated by the product of the probabilities of the individual transformations, assuming they tend to be independent,[3] giving us:

$$P(t|s) \propto P(t) * \prod_{1\ldots k} P(\Delta_k)$$

This form of the equation enables us to directly evaluate the relative probabilities of alternative mappings. The key to doing this accurately lies in determining the probabilities of the individual transformations, such as synonyms, abbreviations, spelling mistakes, etc.

We determine these probabilities using supervised machine learning. The algorithm is "supervised" because it starts with labeled examples. For instance, the hospital staff provides training translations, such as "Beta-HCG" matches "Beta human chorionic gonadotropin urine qualitative" from which the algorithm learns how to perform the matching. Each training example is then automatically converted into a transformation graph, such as the one in Figure 2, which shows how different pieces of the terms map to one another (e.g., which transformations apply across the various terms).

Using our example of "Beta-HCG" a transformation graph might show that "Beta" maps to " Beta" via an "equals" transformation and "HCG" maps to "human chorionic gonadotropin" via an "acronym." However, there are usually other transformation graphs that represent other, possibly worse, translations. For instance, "Beta-HCG" could also map to "Beta blocker" since the first token of "Beta" applies with an "equals" transformation. However, the rest of the tokens (blocker and HCG) are unaccounted for. Therefore, the goal is to learn the most likely transformation graphs levering the labeled data. To do this, we employ a heuristic approach that determines the most likely combination of transformations for each pair in the training data (Minton et al. 2005). This model is then stored and use to compute the probabilities at runtime, determining the translations of health terms by assigning the most likely transformations.

Compiling Health Language Transformations

As we describe above, our approach hinges upon uncovering the various types of transformations that may occur across different health language terminologies. However,

[3]We note, we make an assumption of independence based upon the Naive Bayes assumption. That is, we assume that if we encounter a misspelling and a synonym, there is no reason, a priori, to believe that the occurrence of one influences the occurrence of the other. This allows us to keep our combinations order invariant, which helps make the search more efficient. That is, we only have to consider that a misspelling and a synonym occurred, and we do not care about the order in which they appeared.

health terminologies are quite complex, and often traditional transformations used to match items such as product names or people names (e.g., acronyms, equal-tokens, prefixes) are not sufficient to capture all of the relationships across medical terms. Therefore, we augment the set of traditional transformations with medical-domain specific transformations that we can learn directly from various health related data sources. Here we briefly describe some of our initial approaches to mining these transformations, which are crucial for our approach, yet costly to develop by hand (hence the need for data mining approaches).

For clarity in describing our experimental results below, we group various transformations we used by the approach that generates them for use in Health E-Match. For instance, as we will describe, we have a few different ways in which to uncover synonyms that are useful for mapping health languages, so we name them to make it clear.

Medical Domain Knowledge: To build our first set of transformations we harvested and applied knowledge from a number of outside information sources such as SNOMED, RxNORM, the FDA Product Database, and MESH. By ingesting these sources we produced specialized transformations for the medical domain such as anatomical synonyms, transformations mapping the generic names to brand names for drugs, medical abbreviations, etc. This set represents a large and comprehensive source of general medical information for our approach, which includes commonly used transformations such as synonyms, IS-A relationships, subconcepts, acronyms, and other standard ontological relations. This module is capable of efficiently translating arbitrary external data sources represented in various formats (including user-provided lists) into the appropriate Health E-Match transformations.

Shallow Medical Parser: In some cases, it is helpful to parse the source and target items and then reason about their ontological/linguistic constituents. For instance, Figure 3 shows two medication descriptions, broken down into constituent parts of Dose, Drug name, Frequency and Route. By parsing these descriptions into their constituents, we can then describe the similarity/differences between these descriptions in terms of the constituents, as shown on the right of the figure. To incorporate this type of transformation, we use a look-up parser based upon the sources used in the Medical Domain Knowledge to tag items such as medications or procedures. As an example, it would tag the token "Tylenol" as a "Drug."

Instance-based Synonyms: This approach leverages instance-based learning (Russell and Norvig 2003) to uncover unusual term substitutions. In this procedure, all of the historic mappings for a given translation are used to propose possible wholesale substitutions, where an entire data item is substituted for another entire data item. This variant appears to be particularly useful discovering idiosyncratic hospital jargon that is not well captured by standard sets of medical ontologies and databases.

Synonym Miner: Our final approach allows the system to learn synonyms automatically by analyzing matched data items. (We use the term synonym generically here - in actuality, this algorithm produces a variety of word substitutions,

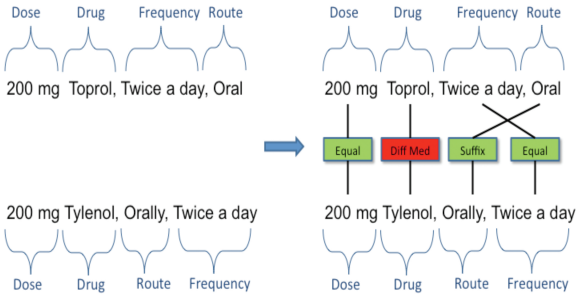

Figure 3: Parsing and matching

including abbreviations, acronyms, proper synonyms, etc.) The intuition is that this approach finds pairs of words that appear to fulfill the same role in matching data items. For example, consider the matching items about a cat-scan of the abdomen and pelvis region: CT ABD PELVIS URO and CT ABDOMEN PELVIS UROGRPAHY. Since the words CT and PELVIS are in common across both, this results in two suggested synonyms: ABD = ABDOMEN, and URO = UROGRAPHY. To run this procedure, we align all pairs of known translations (e.g,. from training data, or even from corpus' that have variants of the same item[4]), and then find term substitutions that differ where many of the other words are exactly the same. To minimize false positives, we must see significant support for that pair (for instance, we see that ABD aligns with ABDOMEN in at least some high percentage of the examples where this occurs, such as 90% of the time).

Health E-Match Architecture

With these definitions in place, we now describe the overall architecture for Health E-Match, shown in Figure 4. Overall, the Health E-Match system is composed of three, higher level components. There is the "Learning" component (shown on the left of the figure), the "Matching" component (shown in the middle of the figure), and the "Data Management" component (on the right). The Learning component incorporates the various machine-learning approaches discussed above. It's goal is to learn the probabilities of transformations, as well as support the loading and discovery of the various transformations (see previous subsection). Once it learns the different transformations and their probabilities of occurring, it stores this information in the Matching component. The Matching does the actual mapping between a data item s from a source language S and the various data items $t \in T$, the target language.

A detailed description of matching is beyond the scope of this paper, as our intention here is to introduce the overall Health E-Match framework. However, at a high level, the matching process breaks into two steps (each building upon our previous work). For the first step of matching, only the most likely *candidate* matches are proposed by the system. This is known as "blocking" and is meant to improve the

[4]The above example, for instance, comes from the RadLex playbook.

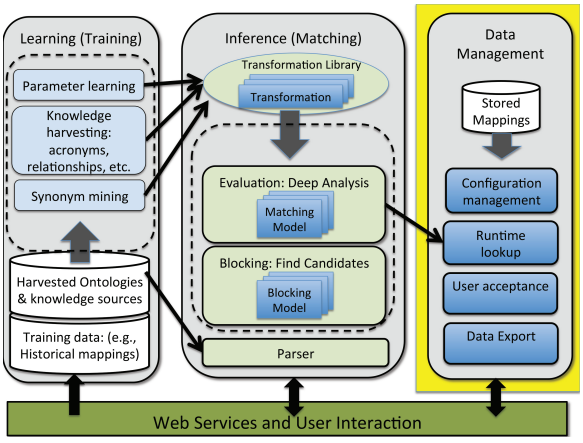

Figure 4: Health E-Match architecture

efficiency of matching by limiting the number of potential translations to consider to only those that are most likely to be correct. For example, if an incoming term is "Beta human chorionic gonadotropin urine qualitative," it might limit possible matches to only those terms that contain the word "beta" or "qual" or "urine" in them. Here we leverage previous work where we showed how the rules that define blocking (e.g., must have the token "beta" or prefix "qual") can be learned from training data (Michelson and Knoblock 2006). However, note that blocking also includes potentially incorrect matches such as "beta blocker" and "urine pouch." The goal of blocking, therefore, is not to find all of the correct matches, but to limit the size of the potential matches that must be examined. So, blocking is really an approximation that improves efficiency. Once the candidate matches are found, they are examined in detail during the second step which we call "evaluation" (as shown in Figure 4). The evaluation step computes the probability that each candidate matches the input phrase using the Bayes-approximation based upon the transformations, as described above.

Finally, once matches are made, they are passed to the Data Management component. This component is used to interact with actual users, storing their mappings (and supporting human-in-the-loop approval for real-world use). The workflow is such that the system proposes a top match (and potentially a few others, in case it is wrong) and a human administrator then oversees all of the matches, approving those that are correct and correcting those that are not. These are then stored for real-time lookup, so that at runtime, the translations can happen on-the-fly. Since this data management component is also beyond the scope of this paper, we put a yellow box around it to highlight that it is not within the scope of this discussion.

Experiments and Results

We tested the Health E-Match system with large amounts of real-world data to verify two claims. First, we verify the accuracy of our approach to semantic interoperability using a large and challenging data set for translation. Second, given the (potential) cost involved in training a new system (since truth labels need to be assigned), we demonstrate how an "off-the-shelf" model performs and how it can be customized to perform health language translation in a totally new setting (mirroring the process of local-to-local mappings). This illustrates the potential for a universal application of health-language translation where we can map local health terminologies to other local health terminologies.

Our experimental data was provided by two large, leading healthcare companies. Company A is a large health-data and integration provider. Company B is a radiology-device and software company. Both companies share a similar use-case for their data: translating data from multiple partners/customers into their own internal language, in order to link incoming data to their systems.

Company A provided data consisting of hundreds of thousands of translations, composed of "hospital orders" (medications, lab tests, diets, consultation requests, etc.) from hundreds of hospitals that were each mapped to Company As internal terminology by their own medical language experts. For our experiments here we focused on a data set of 25,000 translation pairs coming from 10 different hospitals, representing some of the most challenging translations. As an example, Company A provided the translation "D5-0.9% Sodium Chloride with Potassium Chloride 20 mEq/L IV," which maps to "KCL 20MEQ in D5W-0.9% NACL-LTR 20 MEQ/1000 ML IVSL." Notice the word-to-chemical substitutions (KCL for Potassium Chloride), the numeric equivalence, and even the varying representation for the route (IV versus IVSL).

The Company B data contains slightly more than 5,000 hospital-provided descriptions of radiology procedures mapped to the Radiological Society of North America's "RadLex Playbook," a standard terminology for describing radiology studies. As an example in the Company B data, an input is "Thoraxô2_PulmAngioCTA Radia" which maps to the following terminology from the RadLex Playbook, "CT CHST ANGIO PULM ARTS W IVCON." Again, this is a challenging, but real-world, task for health language translation.

Our first experiment focuses on the larger Company A data, and demonstrates the overall effectiveness of our approach, demonstrating the incremental improvements that come from including different transformations one at a time. Our experiments measure and report "Precision@K." This measurement tracks how often the correct translation is found within the systems top-K ranked choices (ranked by score). For instance, a Precision@10 score of 90% indicates that 90 percent of the time, the system ranked the best translation (determined by a domain expert) as one of its top 10 choices. We report Precision@1, which reflects how accurate the system is running fully autonomously. We also report Precision@5 and Precision@10, which reflect how the system would perform in a human-in-the-loop scenario (e..g, where the system suggests the top matches and an "administrator" then picks the best from the list). In discussions, Company A felt that Precision@10 was the best choice for measuring performance, because it directly correlates to the reduction in manual labor that would accrue from their use of the system. In Company As use case, a human expert al-

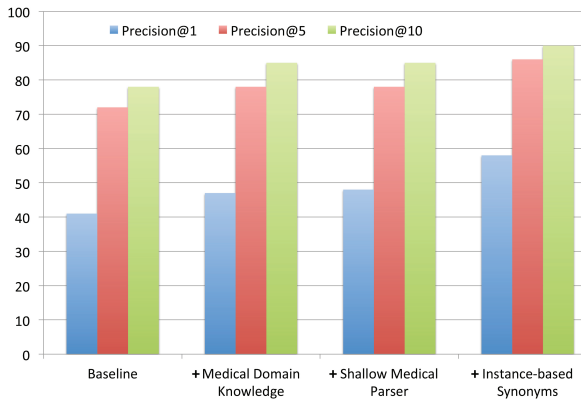

Figure 5: Translation Results: Company A data

ways needs to be the final arbiter of a mapping, and a human can almost instantaneously scan the top 10 results (but more than 10 quickly becomes burdensome).

In this experiment, as we mention above, the Company A data set contains aggregated hospital orders from 10 different hospital systems. Therefore, rather than run a more artificial cross-fold validation experiment (the traditional choice) we instead ran this experiment using a hold-one-out procedure. This means that we train on data (build our model) using nine of the hospital systems, and then test the model on data from the tenth. A different hospital is held out across each of the ten trials, and we can then average our reported results. This allow us to compute how well the model performs on average, and also demonstrates how well the model generalizes to previously unseen data, since the hospital systems are independent from one another (that is, there is no reason, a priori, to expect that two hospitals should share the exact same mappings). Finally, this mimics the process of integrating a new hospital system's data into a previously constructed data integration system, which is the end goal of this language translation research.

As mentioned above, this experiment isolates the contributions from different translation components. Therefore, we begin by training and reporting results using a baseline algorithm. The baseline is our out-of-the-box probabilistic translator. It employs our language model methodology, but only includes generic transformations such as "equal words," "misspelling," etc. We then layered on different types of advanced transformations (e.g., medical knowledge, mined synonyms, etc.), stacking one after another (that is, we include one transformation with the baseline, then two with the baseline, etc.). Since the transformations are constructed and applied in an independent manner, we can isolate the differences by understanding the improvements both before and after adding a specific transformation. However, stacking the transformations also allows us to build a richer and richer model, such that the final model, which includes the most transformations, indicates how well we can perform over the baseline with our full approach (e.g., the set of medical-specific transformations).

The results are shown in Figure 5. Each set of transformations layered upon the previous result is shown in the X-axis, named in correspondence to their description in the section above on compiling the transformations. The main result is that the cumulative result of these learning methods achieves an average Precision@10 of 90.5% across the data sets (the standard deviation is 4.7%). Further, the top suggested translation was correct in 31.8% more of the cases over the baseline. Again, we emphasize that in the hold-one-out scenario we are testing on data completely isolated from the test set (since they occur at separate hospitals).

There are a number of secondary results that also warrant discussion. As the graph shows, adding the Medical Domain Knowledge, which leverages the external sources, gives the model a significant boost. Next, while it is hard to discern from the graph, the Shallow Medical Parser yields the same Precision@5 and Precision@10 results, but boosts the Precision@1 value. This reflects the fact that the parser helps the system do a better job at ranking the top few results, in effect capturing some of the more difficult and nuanced cases. Finally, adding in the instance-based synonyms produces a significant boost across all three metrics. As we stated previously, the instance-based synonyms capture very hospital specific information. Therefore, this result reflects that much of the information that can be leveraged for translation appears to reflect information that is quite idiosyncratic to a hospital not well captured in other ways (such as structured medical knowledge, parsing, etc.).

Our second experiment focuses on whether our technique could be a potential direction toward a more universal approach to health language translation. That is, as we accumulate more knowledge and data over time, is it possible to use our learned models to translate data from new sources? To test this concept, we applied a model trained on nine of the Company A hospitals data to the completely different set of data provided by Company B. Again, as noted above, this experiment aims to map radiology orders, provided by Company B, to their equivalent in the Radlex Playbook, a standardized terminology for radiology. However, we will use a model trained on Company A data to perform the mappings (We note, the Company A data does not include the RadLex Playbook in its Medical Domain Knowledge). The Company A data focused broadly on orders for medications, radiology studies, preparations, communications and discharges across hundreds of hospitals. Meanwhile, the Company B data focuses specifically on radiology orders from its own, disjoint set of radiology departments. As with the Company A experiment, we ran a number of different trials, each building upon the previous with a different addition to the model, and report Precision@K. The results are shown in Figure 6.[5]

The first result is that indeed, the Company A model generalizes to this new data. The Baseline result in the figure reflects the out-of-the-box Company A model applied to this new data, reflecting the potential of applying a pre-built

[5] As we discuss in the next section, the current Synonym Mining component can introduce noise, especially when run on very large data sets, and therefore was not included in this experiment. However, given that the next experiment is smaller, we ran it for the Company B data.

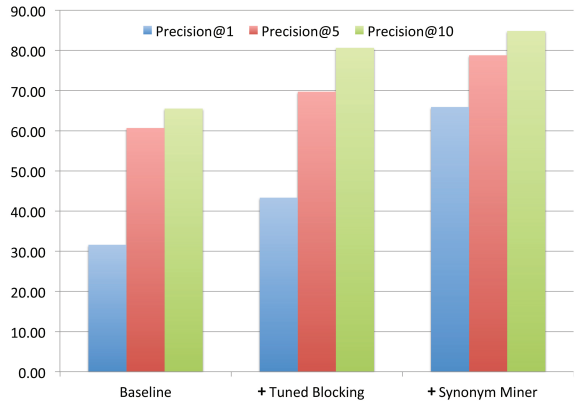

Figure 6: Translation Results: Company B data

model to new a new language translation task. It is interesting to note again that the Company A model was trained on a much more comprehensive data set consisting of orders ranging across a number of areas, not just radiology, while the Company B data is radiology specific. We note, that the Company A model was that trained with all of the transformation components described above.

The next result, "Tuned Blocking" reports the results when we elimintated the blocking algorithm used for Company A's data. As we mentioned, blocking is part of the matching process that attempts to limit the necessary pairs of translations to examine to be only those that are likely to be correct. Blocking is necessary for Company A's data because analyzing all possible pairs of translations is prohibitively expensive (it could number in the billions for Company A). However, we found that the blocking algorithm configured for Company A was actually detrimental to the performance on Company B's data. That is, rather than helping matching overall, the blocking was too restrictive when applied to Company B's data, filtering out too many of true matches when it tried to limit the data to only the potential candidates. Since Company B's data sets are small enough that all pairs could feasibly be evaluated, we turned blocking off in this case and the accuracy improved significantly. Eventually, we hope for Health E-Match to learn to identify these situations (based on training data) so it can determine itself whether blocking is appropriately configured or not.

Building upon the idea of customization, the next model improvement involved mining synonyms directly from the data. We fed the synonym-mining algorithm the Radlex Playbook, where it discovered 144 synonyms by examining variations in the "short name" for a procedure versus the "long name" for a procedure (e.g., "CT ABD" versus "CT ABDOMEN"). The synonym mining is still early work, and as such, can lead to some errors. Therefore, we pruned out roughly 15% of the mined synonyms that seemed incorrect. We note, these types of synonyms are very specific to radiology, and perhaps would not work in a more general context, for instance where one cannot assume that URO is UROGRAPHY, but could mean something else. This is an example of where the targeted customization is beneficial. Again, we note that the Company A model already contained the other transformation components, and so this is the last transformation type to layer onto the model.

Overall, the results illustrate the potential of our approach. We were able to take a model trained on data from Company A, and use it to produce results on data from Company B. Moreover, even though this is early work, the Precision@10 figure (our primary measure of performance) reached 85%, as shown in Figure 6.

Related Work

Given the significance of the interoperability problem, it is unsurprising that there are already a number of terminology mapping services, some of them commercially available. They include Apelons Distributed Terminology System; 3Ms Healthcare Data Dictionary (HDD), and its open access version, HDD Access; and the Regenstrief Institutes RELMA tool, used to map laboratory orders to the LOINC standard terminology. However, to the best of our knowledge, all of these systems map single concepts into fixed, standard dictionaries, and therefore they all suffer from the same issues.

In contrast to our machine learning approach, as we understand, all of these tools employ different heuristics or rules to map into the various terminologies and are therefore not adaptable to user needs or changes in the data. Fundamentally, since they cannot adapt, they cannot be customized, which is a major feature requirement for many customers who want to choose what language to translate into (e.g., their own local language) or who may have data that doesn't fit well with the hand-crafted rules (e.g., one with many customized synonyms). This inflexibility leads to a number of problems that machine learning can address:

- These systems cannot learn how to deal with extremely unique or noisy mappings. Instead the system must be extended with new rules or heuristics for each mapping that fails. In contrast, a machine learning approach can learn from examples how to cover new and interesting cases.
- The system may not be built to deal with data items composed of multiple concepts. Therefore, a term such as "XR Toes Great Left" which involves both a procedure, "X-Ray," and a body part, the left great toe, is generally mapped to "left toe" representations since the system must make a choice about single concepts.
- None of these systems could easily support local-to-local vocabulary mappings. That is, none of them support two-way translation. For these systems, the developers have created the rules and heuristics to map some terminologies to the standards they support, but they dont have a mechanism to support arbitrary health-language to health-language translations. With a machine learning approach, one can build models using the same software, simply by providing different training examples or even customizing a previously built model with new learning algorithms, as we did in our Company B experiment.

In many ways, the progress of health language translation has followed a common course in machine learning. In ap-

plication after application (search engines, language translation, news content analysis) the field has followed a familiar pattern that starts with somewhat brittle systems that rely on rules and heuristics and evolve to systems that learn from the data, which can adapt over time with the data. In this regard, we believe our approach is the next generation technology for translation health languages.

Discussion

There are a number of future research directions we plan to investigate. The first, and potentially most obvious issue, is whether 85% (Company B) to roughly 90% (Company A) accuracy is good enough. We believe indeed this is the case and that Health E-Match is viable, now, given a human-in-the-loop approach where an administrator oversees the matches and chooses the best proposed by the system. Of course, we would prefer a fully autonomous system, where the administrator just supervises ("yes or no") the proposed match, rather than having to both supervise and pick the best one from a list. Yet, although not yet deployed, we received positive feedback from users at Company A on our approach. They found it yielded significant cost savings since the humans didn't have to take as much time to find the matches, and that the approach was indeed helpful and useful. Further, we could potentially extend the learning capabilities based upon feedback from the administrators. Essentially, each "correction" by an administrator (e.g., picking a match in the top-X but not ranked number 1) provides valuable data for the system to learn from over time.

The human-in-the-loop aspect raises an interesting, yet anecdotal, facet we plan to investigate further. Specifically, although Health E-Match only achieves 85% accuracy on new data, we are certain that humans don't achieve 100% themselves. Some of the many thousands of labeled data we received had errors in them, raising this prospect (though not enough to be quantified). In fact, in the future we would like to run a labeling experiment where we can measure the Kappa agreement (Carletta 1996) between various people assigning matches to the terminology.[6] This would help us understand the limits of human ability in this task, and allow for us to judge whether the system was performing at a comparable level to people.

Another area for improvement is in discovering and incorporating more transformations. The mining of synonyms from different data sources is a rich and interesting pursuit and should provide significant improvements for a system such as ours. Further, there are more complicated transformation types that could yield improvements. For instance, incorporating hierarchical relationships, such as those encoded by ontologies could yield improved inferences. Finally, there are issues when multiple transformations are applied to a single field. For instance, imagine a field such as "chust" which both a misspelling of "chest" and a synonym for "thoracic." In this case, the system would need to recognize and apply both transformations at the same time (e.g., first recognize that "chust" is a misspelling of "chest" and then realizing it should be matched to "thoracic.")

[6] A Kappa score encapsulates how well two people agree in similar data mark-up tasks.

Conclusion

In this paper we presented Health E-Match, an approach for semantic interoperability across health providers. The system uses a language-modeling approach to *translate* the data fields across systems, and supports mapping noisy data, local-to-local (versus local-to-standards) mapping, etc. It is flexible and accurate, and as we demonstrated in our experiments begins to address the problem of mapping local-to-local data (e.g., using a model trained on completely different health data to map another source's data).

Acknowledgements

This material is based upon work supported by the United States Army Medical Research and Materiel Command under Contract No. W81XWH-13-C-0082 and work supported by the National Science Foundation under Grant No. IIP-1330223.

References

Carletta, J. 1996. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.* 22(2):249–254.

Koehn, P. 2010. *Statistical Machine Translation*. New York, NY, USA: Cambridge University Press, 1st edition.

Michelson, M., and Knoblock, C. A. 2006. Learning blocking schemes for record linkage. In *Proceedings of the 21st National Conference on Artificial Intelligence*.

Minton, S. N.; Nanjo, C.; Knoblock, C. A.; Michalowski, M.; and Michelson, M. 2005. A heterogeneous field matching method for record linkage. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM-05)*.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)* 34(1):1–47.