

LINKING GENES IN LOCUSLINK/ENTREZ GENE TO MEDLINE CITATIONS WITH LINGPIPE*

BRECK BALDWIN AND BOB CARPENTER

Alias-i, Inc.
181 N. 11th St, #401
Brooklyn, NY 11211 USA
E-mail: breck@alias-i.com

This paper demonstrates that the human curated citations associated with genes in Entrez Gene (formerly LocusLink) provide an accurate method for tracking gene references through the biomedical literature as represented by MEDLINE. We show how the Entrez Gene citations for a gene can be used to build character language-model-based classifiers that picks out MEDLINE citations about that gene. This problem is harder than it may appear given the range of overlapping aliases and contexts. We use the language modeling and classification modules of LingPipe, a natural language processing toolkit distributed with source code.

1. Introduction

An interesting and important challenge in bioinformatics concerns linking database information about genes, proteins and other entities of interest with free form text information as found in sources like MEDLINE abstracts (<http://www.ncbi.nlm.nih.gov/entrez>) and full research articles¹³. This paper provides (1) an analysis of what information in databases can contribute to automatically linking to MEDLINE abstracts, and (2) the evaluation of language model classifiers for using human curated abstracts to train a recognizer for linking genes to MEDLINE abstracts about them.

2. A Basic Analysis of the Scope of the Problem

The Entrez Gene database, previously known as LocusLink, contains approximately 20,000 entries for human genes. Below is an excerpt of some of the relevant fields for this study:

LOCUSID: 12

*This work is supported by NIH Grant 1 R43 RR020259-01.

2

```
...
OFFICIAL_SYMBOL: SERPINA3
OFFICIAL_GENE_NAME: serine (or cysteine) proteinase inhibitor, clade A
(alpha-1 antiproteinase, antitrypsin), member 3
ALIAS_SYMBOL: ACT
ALIAS_SYMBOL: AACT
...
SUMMARY: Summary: The protein encoded by this gene is a plasma
protease inhibitor and member of the serine protease inhibitor class...
...
PMID: 15014966,14718574,14702039,12709365,12600202,12477932,...
```

The OFFICIAL_SYMBOL, OFFICIAL_GENE_NAME and ALIAS_SYMBOL fields correspond to alternative names for a given gene. The PMID field lists human curated MEDLINE citations which reference the gene. The SUMMARY is an additional text source that is perhaps useful in identifying further documents that mention the gene. The goal of this research is to add further machine recognized MEDLINE citations which reference a given gene in Entrez Gene.

2.1. Ambiguity and Resolution in Keyword Fields

To better understand the issues in linking Entrez Gene to MEDLINE, we analyzed the relationship between the various keyword fields in Entrez Gene and the curated citations.

In Entrez Gene, there are 19,970 database entries for human genes. These genes are associated with a total of 72,048 distinct names in the OFFICIAL_SYMBOL, OFFICIAL_GENE_NAME and ALIAS_SYMBOL fields. Of the 90,391 human curated citations for the genes, only 49,592 (54.8%) contain an exact string match from any of the above fields. This leaves 40,799 (45.2%) citations without an exact string match from the database record.

In examining the citations that didn't match we found that often subtleties of formatting, hyphenation or capitalization were responsible for the lack of an exact string match. Below is an example of a title/abstract from a MEDLINE citation that fails to match any of the aliases MT1, MTP, MT1B for Entrez Gene ID 4490.

Title: Human metallothionein genes: structure of the functional locus at 16q13.

Abstract: The functional human metallothionein (MT) genes are located on chromosome 16q13. We have physically mapped the functional human MT locus by isolation and restriction digest mapping of cloned DNA. The mapped region contains all sequences on chromosome 16 that hybridize

to metallothionein gene probes and comprises 14 tightly linked MT genes, 6 of which have not been previously described. This analysis defines the genetic limits of metallothionein functional diversity in the human genome.

The aliases are too detailed to match the literal instance in the abstract of MT, presumably world knowledge on the part of the curator contributed to identifying that this abstract was relevant to the gene in question. Other examples of near hits from other abstracts for Entrez Gene ID 4490 are: (MT) IB gene, hMT-IB, MT-1, metallothionein-1, metallothionein-1g, MT-I/II, and hMT-IA.

While work has been done to help make keyword matching fuzzier through a number of means mostly involving extensions to weighted edit distance^{10,11}; without further refinement, these approaches increase recall at the expense of precision.

2.2. Whole Document Contextual Classification

We believe techniques for finding matches of aliases are complementary to the techniques discussed in this paper for disambiguating them. General systems for coreference resolution tend to employ a mixture of word-internal (spelling, pronunciation, dictionary based) and contextual features (keyword in context).³ Contextual approaches using words around the term in question have been widely adopted for word sense disambiguation^{2,6} and for noun phrase coreference resolution. For instance, we applied a TF/IDF classifier to disambiguate 38 different people named “John Smith” in two years of *New York Times* articles using sentential contexts.^{1,7}

3. Experimental Setup and System

Our evaluation methodology attempts to bootstrap off of the existing human curated gene to citations mapping in Entrez Gene. These known positive mappings form an excellent foundation for training but are a bit more problematic for testing because there are no data indicating that a particular citation does *not* contain mention of the gene. Consequently we evaluate positive instances in a different fashion than assumed negative instances.

3.1. Collection and Evaluation of Positive Citations

Positive citations can be read off of the PMID field on the database entries. The title and abstract are then available for training. In addition the the the

OFFICIAL_SYMBOL, OFFICIAL_GENE_NAME and ALIAS_SYMBOL SUMMARY fields are used from the database entry for the gene. Evaluation is modeled after the evaluation in the AZuRE system¹⁵ which featured a leave one out (LOO) evaluation methodology for mapping genes to citations. The LOO evaluation technique is the limiting case of cross validation techniques where the number of held out cases is equal to 1. Given 20 known documents which mention a gene by human curation, the LOO evaluation would train on 19 documents, and evaluate on the remaining document returning an entropy estimate for how good a fit the held out document is to the 19 training documents. Iterating through all 20 documents in this fashion yields 20 data points from which performance may be estimated.

3.2. Collection and Evaluation of Negative Citations

Collecting negative citations requires a bit more artifice. We again follow the lead of the AZuRE system by assuming that if two genes share an alias then the MEDLINE citations for gene1 do *not* contain mention of gene2 and vice versa. This way we can assemble assumed negative data for evaluation.

The evaluation is a bit different from the positive citations. It does not make sense to use the LOO methodology because our classifiers only train on the positive data—if we had negative models like the AZuRE system, then we would be obliged to do a LOO evaluation on negative citations as well. To evaluate the negative cases, we first train the positive model on all positive citations and collect the models estimates for each negative citation. In the end we will have two lists of positive and negative estimates on a per-citation level.

System output for a gene looks like:

```
GeneID  PositiveCites  NegCites  FilteredCites
6677      8             7           1
Aliases:  HYAL1,SPAG15,SPAM1,HYA1,PH-20,HYAL3,MGC26532,PH20
BestF:   .94
Entropy type  Prec Recall  F   NumChars  CitePMID
-1.93, pos   1.00  .12  .22  958  8234258
-1.98, pos   1.00  .25  .40  1138  8195297
-2.01, pos   1.00  .38  .55  2272  11731269
-2.14, pos   1.00  .50  .67  2117  2269661
-2.23, pos   1.00  .62  .77  648  8282124
-2.40, pos   1.00  .75  .86  1228  11114590
-2.43, neg   .86  .75  .80  1235  10493834
-2.50, pos   .88  .88  .88  1419  8575780
-2.66, pos   .89  1.00  .94  1689  11731267
-3.05, neg   .80  1.00  .89  1306  12684632
-3.08, neg   .73  1.00  .84  927  10644992
```

-3.15, neg	.67	1.00	.80	1860	11929860
-3.30, neg	.62	1.00	.76	2015	10339581
-3.43, neg	.57	1.00	.73	1760	12084718
-3.75, neg	.53	1.00	.70	2913	11085536

The output shows sorted estimates for both positive and negative citations evaluated against the positive model as discussed above. Where TP, FP, FN are true positives, false positives and false negatives respectively, *precision* is the percentage of positive classifications that were correct, $P = TP/(TP + FP)$, *recall* is the percentage of positive cases that were correct, $R = TP/(TP + FN)$. The F_1 *measure* is their geometric average, $F_1 = 2 \cdot P \cdot R/(P + R)$. The post-hoc (aka oracle) best F-measure operating point is a widely applied single number used to measure ranked classification performance.

3.3. Evaluation Against Entre Gene

There are 2,094 aliases in Entrez Gene which are mentioned in two or more gene entries. Each ambiguous alias is evaluated by iterating through the n gene entries that mention the alias, selecting one entry as the source of positive citations and the remaining $n - 1$ entries as the sources for negative citations. A consequence of this is that the positive and negative citations are of equal size at the end of the evaluation with considerable variation on their respective sizes for ambiguous aliases. There must be at least 2 positive citations or the gene is not evaluated as a positive instance but it will be used as a source for negative citations. We follow the AZuRE convention of excluding citations if more than 20 genes refer to the citation because they tend to involve large scale sequencing projects.

We run the process above for each gene. This allows cumulative results macro-averaged (by gene) F_1 , $\sum_{g \in G} F_1(g)/|G|$, and micro-averaged F (by citation), $\sum_{g \in G} F_1(g) \cdot |g|/\sum_{g \in G} |g|$, where g is a gene in the set G of genes, $F_1(g)$ is the F_1 measure for the gene g , $|G|$ is the number of genes, and $|g|$ is the number of citations for gene g .

3.4. Relation of Evaluation to End User Needs

Linking Entrez Gene entries with MEDLINE citations can be used two ways. First, it may be used to find more citations for a given gene given a few examples of documents or abstract mentioning that gene. For this application, a filter that would preselect the documents with some approximate match of an alias would also be useful. Conversely, our approach

may be used to classify a document (abstract, full paper, etc.) with respect to the genes to which it refers. Note that the results we provide are ranked, and thus support standard paging of results for information retrieval browsing.

The F-measures reported give an indication of the expected quality of the retrieved citations. Despite our use of the ambiguous portion of Entrez Gene, our results apply to the more general unambiguous linking problem because 45 percent of all citations fail to have an exact string match. One can think of the no-keyword match case as similar to the ambiguous case because the known aliases are unable to help make the correct distinction.

4. Binary Classification by Thresholded Language Models

A *language model* provides a probability mapping over sequences of symbols. In this work, we consider language models over sequences of characters and over single tokens. As previously discussed, titles and abstracts are concatenated to form the text of a MEDLINE citation. This allows language models to be used to score citations.

4.1. Character Language Models

Character-level modeling for information retrieval and classification is not a new concept, having been used as early as 1974 for language classification¹⁶, with regular appearances since then^{5,17,14}.

We use the standard form of n -gram character language models. The chain rule, $P(c_1, \dots, c_n) = P(c_1)P(c_2 | c_1) \cdots P(c_n | c_1, \dots, c_{n-1})$, allows us to reduce the language modeling problem to that of estimating the probability of a character given the sequence of characters preceding it. We make the standard n -gram approximation that only the previous $n - 1$ characters are used, $P(c_k | c_1, \dots, c_{k-1}) \approx P(c_k | c_{k-n+1}, \dots, c_{k-1})$.

We assume that characters are encoded in unicode for generality; MEDLINE goes beyond ASCII, as do the full papers. For characters c, d and character sequence X , we define interpolative smoothing of maximum likelihood estimators:

$$P(c|dX) = \lambda(dX)P_{\text{ML}}(c|dX) + (1 - \lambda(dX))P(c|X)$$

$$P(c) = \lambda()P_{\text{ML}}(c) + (1 - \lambda())(1/|\text{Char}|)$$

The maximum likelihood estimators are given by:

$$P_{\text{ML}}(c|X) = \text{count}(Xc) / \text{extCount}(X)$$

where $\text{count}(X)$ is the number of times the sequence X was observed in the training data and $\text{extCount}(X)$ is the number of single-symbol extensions of X observed:

$$\text{extCount}(X) = \sum_{c \in \text{Char}} \text{count}(Xc).$$

Our interpolation ratios are determined using a parametric form of Witten-Bell smoothing based on a single hyperparameter K :

$$\lambda(X) = \frac{\text{extCount}(X)}{\text{extCount}(X) + K \cdot \text{numExts}(X)}$$

where $\text{numExts}(X) = |\{c | \text{count}(Xc) > 0\}|$ is the number of different symbols observed following the sequence X in the training data.

Using suffix-tree style speedups⁸, character 8-gram models train at roughly 200 thousand characters a second and, after compilation, execute at roughly 2 million characters/second on standard desktop PCs.⁴ Only one model is required per category. This provides single-gene classification speeds on the order of 1000 MEDLINE citations/second.

4.2. Naive Bayes with Unigram Token Language Models

Text is more typically modeled at the token level for classification purposes. To this end, we assume a *tokenizer* to break down sequences of characters into a sequence of tokens, each consisting of a non-empty character sequence. We used LingPipe's `IndoEuropeanTokenizerFactory`, which tokenizes at a fairly fine level, for example, breaking "SIL-1" into three tokens, "SIL", "-", and "1", but leaving numbers such as "3.14" together.

To contrast with our character language models and provide a naive Bayes baseline, we ran a test using token unigram models. These are smoothed just like the character language models.

4.3. Binary Classification by Threshold

Given a language model for a class, classification proceeds by setting a threshold on the per-character sample cross-entropy rate. Cross-entropy for a sample text is just the log (base 2) probability of the sample in the model divided by the number of characters. More formally, for a model language model p , the cross-entropy rate of the characters c_1, \dots, c_n is defined to be $H(c_1, \dots, c_n; p) = \log_2 p(c_1, \dots, c_n)/n$.

5. Experiments

We established a baseline system with a Naive Bayes unigram token model and 2 and 3-gram token models, then we sought to establish a reasonable character n -gram size with character language models. In Figure 1 we see results for various classes of language models and ngram sizes. The Naive Bayes baseline performs at 83 percent on both measures and the increased context of token bigrams raises the performance another 3-4 points. The character language models are the best performers with top results with 7-grams. All results except the last row are reported on a 10 percent sample of the data.

NGram	Type	Cite Average F	Gene Average F
1	Naive Bayes	.83	.83
2	Token	.86	.87
3	Token	.86	.87
4	Char	.89	.86
5	Char	.91	.87
6	Char	.91	.87
7	Char	.92	.87
8	Char	.92	.87
9	Char	.91	.87
10	Char	.91	.87
7	Char	.93	.92

Figure 1. Various Language Models for Establishing Type and N-gram size

The top performing model is 7-gram character models which we then ran over all the ambiguous genes available in Entrez Gene for a total of 80,081 citations being evaluated. This is shown in gray on the last row. We speculate that the increased performance is due to a bit of flexibility that the 7-character windows gives moving over data which tokenized models do not have. The input to a 7-gram model would break `...linked metallothionein genes ...` into “ked metal”, “ed metall”, “d metallo”, “metallot”. This last 7-gram would match one of the 7-gram representations of “metallothionen” (it is missing the final “i”).

A more detailed breakdown of the final system performance is shown in Figure 2 as a scatter plot analysis of training set size and F-measure. Please note that data points have been jittered to prevent occlusion leading to values greater than 1. Clearly size of training set has a strong influence on

quality of recognition

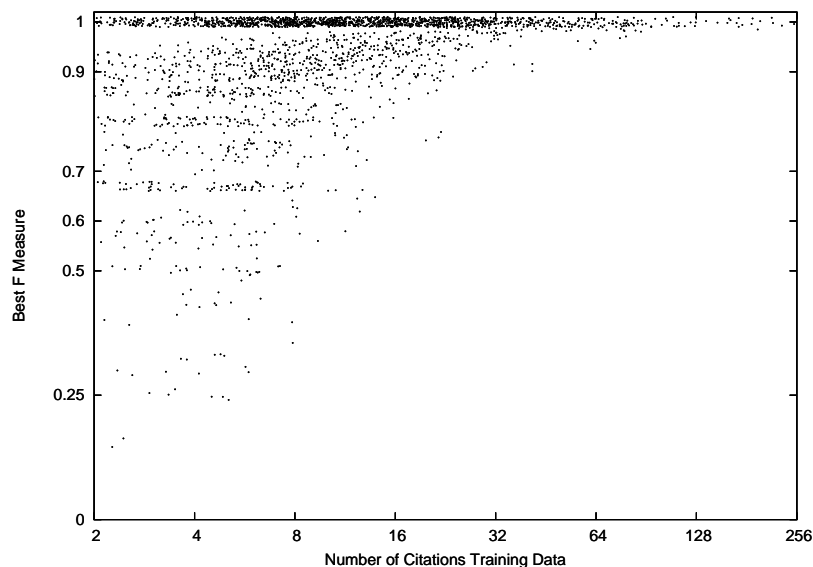


Figure 2. F-Measure versus Number of Positive Training Documents

6. Comparison to AZuRE

The AZuRE system used a hierarchical classification scheme that first determined whether abstracts were about genes at all and then constructed individual classifiers for each gene for evaluation. They used ReelTwo's weighted confidence classifier.¹⁵ This is a Bayesian classifier requiring a positive model $P_{+g}(C|g)$ and negative model $P_{-g}(C)$ and negative model. The negative not-this-gene models were trained on confusable abstracts as well as a random sample from the output of the gene/non-gene classifier above. They balanced training data so that the category probabilities for the positive and negative categories were equal, and the system amounted to choosing either a positive or negative outcome depending on which of P_{+g} and P_{-g} had a higher probability estimate.

We implemented AZuRE's Bayesian approach using our own language model classifiers and found that size of the models was dominating the best fit. The AZuRE authors mention having to balance the two models by various normalizing factors. Our rejection threshold approach arose

because we noticed that a simple threshold on the positive models yielded excellent results as shown in the above experiments.

Additionally, the AZuRE system had a purpose-built tokenizer for Medline abstracts which we did not use in our character language models. This greatly simplified our system since developing and maintaining tokenizers can be troublesome.

6.1. *AZuRE Comparison Results*

In Figure 3 AZuRE system results are in white, LingPipe in grey on the following line. The row “ID” corresponds to the Entrez Gene ID, “Alias” to the ambiguous alias that is referenced by more than one LocusLink record, “G” to the number of curated known gene Medline abstracts in LocusLink, “CG” to the number of found confusable genes from other LocusLink records, “NG” to the number of negative genes (CG + randomly selected abstracts) “CF” refers to the number of abstracts filtered for being referenced by too many LocusLink entries (n is greater than 20) and “F” to the F-measure for the relevant system output. Note that the CG value was derived for the AZuRE results by subtracting G from NG, but this results in a ‘-1’ value in one case so this assumption may be mistaken even though it is supported by the text of the AZuRE paper.

In the interests of transparency we provide counts for positive and negative abstract counts from both the AZuRE paper and our system output. Counts often differ which is expected since we are working with different versions of MEDLINE and Entrez Gene, but we have done our best to align our data with AZuRE’s. For the evaluation in Figure 3 (but not the earlier results), we followed AZuRE in also using SwissProt as an additional source of MEDLINE citations for a given gene.

If any conclusions were to be drawn from these results it would be that the approach taken here is simpler and achieves similar results.

7. Conclusions

We presented promising initial results for linking genes referred to by Entrez Gene to MEDLINE abstracts. The distinguishing feature of our approach is the robust underlying classification scheme based on character language models. This eliminates the need for tokenization, a non-trivial problem for text processing applications particularly for bioinformatics.¹³ In addition, the underlying implementation runs at speeds of 2 million characters/sec on standard desktop hardware.

The source code for this paper is available as a LingPipe demo from at <http://www.alias-i.com/lingpipe>.

References

1. Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the Association for Computational Linguistics*, pages 79–85, 1998.
2. Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. Word-sense disambiguation using statistical methods. pages 264–270, 1991.
3. Claire Cardie and Kiri Wagstaff. Noun phrase coreference as clustering. In *Proceedings of EMNLP/WVLC*, pages 82–89, 1999.
4. Bob Carpenter. Scaling high-order character language models to gigabytes. In *Association for Computational Linguistics Workshop on Software*, 2005.
5. W. B. Cavner and J. M. Trenkle. N-gram based text categorization. 1994.
6. William Gale, Kenneth Church, and David Yarowsky. A method for disambiguating word senses in a language corpus. *Computers and the Humanities*, 26:415–439, 1992.
7. Chung Heong Gooi and James Allan. Cross-document coreference on a large scale corpus. In *Proceedings of the North American Conference on Computational Linguistics*, 2004.
8. Dan Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
9. Dan Klein, Joseph Smarr, Huy Nguyen, , and Christopher D. Manning. Named entity recognition with character-level models. In *Proceedings the 7th ConNLL*, pages 180–183, 2003.
10. Zhenzhen Kou, William W. Cohen, and Robert F. Murphy. High-recall protein entity recognition using a dictionary. In *Proceedings of ISMB-05*, 2005.
11. M. A. Krauthammer, A. Rzhetsky, P. Morozov, and C. Friedman.
12. John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of ACM SIGIR*, 2001.
13. Alexander Yeh Lynette Hirschman, Alex Morgan. Rutabaga by any other name: Extracting biological names. *Journal of Biomedical Informatics*, 35, 2002.
14. Fuchun Peng, Dale Schuurmans, and Shaojun Wang. Language and task independent text categorization with simple language models. 2003.
15. Raf M. Podowski, John G. Cleary, Nicholas T. Goncharoff, Gregory Amoutzias, and William S. Hayes. Azure, a scalable system for automated term disambiguation of gene and protein names. In *Proceedings of the 2004 IEEE Computation Systems Bioinformatics Conference(CSB 2004)*, 2004.
16. M. D. Rau. Language identification by statistical analysis. Master’s thesis, 1974.
17. William J. Teahan. Text classification and segmentation using minimum cross-entropy. In *Proceeding of RIAO 2000*, 2000.

ID	Alias	G	CG	NG	CF	F
12	ACT	30	10	40	–	.97
		23	4	4	10	1.00
54	TRAP	18	121	139	–	.94
		12	88	88	15	1.00
231	AR	19	259	278	–	.90
		14	153	153	11	.85
367	AR	244	32	276	–	.98
		137	29	29	128	1.00
374	AR	16	259	275	–	.94
		14	151	151	1	.80
434	ASP	8	64	72	–	.88
		6	40	40	12	.92
718	ASP	31	36	67	–	.97
		21	18	18	12	1.00
796	CT	36	12	48	–	.92
		29	10	10	10	.98
847	CAT	24	3	27	–	.96
		19	3	3	1	1.00
948	FAT	35	26	61	–	.94
		37	26	26	9	1.00
1356	CP	26	-1	25	–	.96
		17	0	0	1	1.00
1890	TP	25	34	59	–	.84
		22	30	30	6	1.00
2099	ER	194	2	196	–	.96
		202	6	6	14	1.00
2950	PI	79	66	145	–	.92
		70	39	39	50	1.00
3240	HP	18	12	30	–	.94
		23	8	8	5	1.00
4860	NP	18	23	41	–	.94
		13	19	19	9	.96
5241	PR	45	3	48	–	.96
		49	0	0	1	1.00
5265	PI	67	78	145	–	.91
		39	70	70	36	1.00
6476	SI	7	13	20	–	1.00
		5	13	13	2	.89
7298	TS	32	24	56	–	.97
		–	29	16	16	9

Figure 3. AZuRE results in white; this paper gray