Author's addresses: Diego Marcheggiani, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 56124 Pisa, Italy; Fabrizio Sebastiani, Qatar Computing Research Institute, Qatar Foundation, PO Box 5825, Doha, Qatar. Fabrizio Sebastiani is currently on leave from Consiglio Nazionale delle Ricerche, Italy. The order in which the authors are listed is purely alphabetical; each author has given an equally important contribution to this work.

DOI:http://dx.doi.org/10.1145/0000000.0000000

On the Effects of Low-Quality Training Data on Information Extraction from Clinical Reports

DIEGO MARCHEGGIANI, Consiglio Nazionale delle Ricerche
FABRIZIO SEBASTIANI, Qatar Computing Research Institute

In the last five years there has been a flurry of work on information extraction from clinical documents, i.e., on algorithms capable of extracting, from the informal and unstructured texts that are generated during everyday clinical practice, mentions of concepts relevant to such practice. Most of this literature is about methods based on supervised learning, i.e., methods for training an information extraction system from manually annotated examples. While a lot of work has been devoted to devising learning methods that generate more and more accurate information extractors, no work has been devoted to investigating the effect of the quality of training data on the learning process. Low quality in training data often derives from the fact that the person who has annotated the data is different from the one against whose judgment the automatically annotated data must be evaluated. In this paper we test the impact of such data quality issues on the accuracy of information extraction systems as applied to the clinical domain. We do this by comparing the accuracy deriving from training data annotated by the authoritative coder (i.e., the one who has also annotated the test data, and by whose judgment we must abide), with the accuracy deriving from training data annotated by a different coder. The results indicate that, although the disagreement between the two coders (as measured on the training set) is substantial, the difference is (surprisingly enough) not always statistically significant.

Categories and Subject Descriptors: Information systems [**Information retrieval**]: Retrieval tasks and goals—*Clustering and Classification*; Computing methodologies [**Machine learning**]: Learning paradigms—*Supervised learning*

General Terms: Algorithm, Design, Experimentation, Measurements

Additional Key Words and Phrases: Information Extraction, Annotation quality, Radiology Reports, Medical Reports, Clinical Narratives, Machine Learning

ACM Reference Format:
Diego Marcheggiani and Fabrizio Sebastiani, 2015. On the Effects of Low-Quality Training Data on Information Extraction from Clinical Reports. *ACM J. Data Inform. Quality* V, N, Article A (January YYYY), 17 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

In the last five years there has been a flurry of work (see e.g., [Kelly et al. 2014; Pradhan et al. 2014; Sun et al. 2013; Suominen et al. 2013; Uzuner et al. 2012; Uzuner et al. 2011]) on information extraction from clinical documents, i.e., on algorithms capable of extracting, from the informal and unstructured texts that are generated during everyday clinical practice (e.g., admission reports, radiological reports, discharge summaries, clinical notes), mentions of concepts relevant to such practice. Most of this literature is about methods based on supervised learning, i.e., methods for training an information extraction system from manually annotated examples. While a lot of work has been devoted to devising text representation methods and variants of the aforementioned supervised learning methods that generate more and more accurate information extractors, no work has been devoted to investigating the effects of the quality of training data on the learning process. Issues of quality in the training data may arise for different reasons:

(1) In several organizations it is often the case that the original annotation is performed by coders (a.k.a. “annotators”, or “assessors”) as a part of a daily routine in which fast turnaround, rather than annotation quality, is the main goal of the

coders and/or of the organization. An example is the (increasingly frequent) case in which annotation is performed via crowdsourcing using instruments such as, e.g., Mechanical Turk, CrowdFlower, etc.[1] [Grady and Lease 2010; Snow et al. 2008].

(2) In many organizations it is also the case that annotation work is usually carried out by junior staff (e.g., interns), since having it accomplished by senior employees would make costs soar.

(3) It is often the case that the coders entrusted with the annotation work were not originally involved in designing the tagset (i.e., the set of concepts whose mentions are sought in the documents). As a result, the coders may have a suboptimal understanding of the true meaning of these concepts, or of how their mentions are meant to look like, which may negatively affect the quality of their annotation.

(4) The data used for training the system may sometimes be old or outdated, with the annotations not reflecting the current meaning of the concepts anymore. This is an example of a phenomenon, called *concept drift* [Quiñonero-Candela et al. 2009; Sammut and Harries 2011], which is well known in machine learning.

We may summarize all the cases mentioned above by saying that, should the training data be independently re-annotated by an *authoritative coder* (hereafter indicated as C_α), the resulting annotations would be, to a certain extent, more reliable. We would also be able to precisely measure this difference in reliability, by measuring the *intercoder agreement* (via measures such as Cohen's kappa – see e.g., [Artstein and Poesio 2008; Di Eugenio and Glass 2004]) between the training data Tr as coded by C_α and the training data as coded by whoever else originally annotated them (whom we will call, for simplicity, the *non-authoritative coder* – hereafter indicated as C_β). In the rest of this paper we will take the authoritative coder C_α to be *the coder whose annotations are to be taken as correct*, i.e., considered as the "gold standard". As a consequence we may assume that C_α is the coder who, once the system is trained and deployed, has also the authority to evaluate the accuracy of the automatic annotation (i.e., decide which annotations are correct and which are not)[2]. In this case, intercoder (dis)agreement measures the amount of noise that is introduced in the training data by having them annotated by a coder C_β different from the authoritative coder C_α.

It is natural to expect the accuracy of an information extraction system to be lower if the training data have been annotated by a non-authoritative coder C_β, and higher if they have been annotated by C_α herself. However, note that this is *not* a consequence of the fact that C_α is more experienced, or senior, or reliable, than C_β. Rather, it is a consequence of the fact that standard supervised learning algorithms are based on the assumption that the training set and the test set are identically and independently distributed (the so-called *i.i.d. assumption*), i.e., that both sets are randomly drawn from the *same* distribution. As a result, these algorithms learn to replicate the subjective annotation style of their supervisors, i.e., of those who have annotated the training data. This means that we may expect accuracy to be higher simply when the coder of the training set and the coder of the test set are the *same* person, and to be lower when the two coders are different, irrespective of how experienced, or senior, or reliable, they are. In other words, the very fact that a coder is entrusted with the task of evaluating the automatic annotations (i.e., of annotating the test set) makes this coder *authoritative by definition*. For this reason, the authoritative coder C_α may equivalently be defined as "the coder who has annotated the test set" (or: "the coder

[1] https://www.mturk.com/, http://crowdflower.com/

[2] In some organizations this authoritative coder may well be a fictional entity, e.g., several coders may be equally experienced and thus equally authoritative. However, without loss of generality we will hereafter assume that C_α exists and is unique.

whose judgments we adhere to when evaluating the accuracy of the system"), and the non-authoritative coder C_β may equivalently be defined as "a coder different from the authoritative coder".

The above arguments point to the fact that the impact of training data quality – under its many facets discussed in items (1)-(4) above – on the accuracy of information extraction systems may be measured by

(1) evaluating the accuracy of the system in an *authoritative* setting (i.e., both training and test sets annotated by the authoritative coder C_α), and then
(2) evaluating the loss in accuracy, with respect to the authoritative setting, that derives from working instead in a *non-authoritative* setting (i.e., test set annotated by C_α and training set annotated by a non-authoritative coder C_β)[3].

1.1. Our contribution

In this paper we test the impact of training data quality on the accuracy of information extraction systems as applied to the clinical domain. We do this by testing the accuracy of two state-of the-art systems on a dataset of radiology reports (originally discussed in [Esuli et al. 2013]) in which a portion of the data has independently been annotated by two different experts. In other words, we try to answer the question: "What is the consequence of the fact that my training data are not sterling quality? that the coders who produced them are not entirely dependable? How much am I going to lose in terms of accuracy of the trained system?"

In these experiments we not only test the "pure" authoritative and non-authoritative settings described above, but we also test *partially authoritative* settings, in which increasingly large portions of the training data as annotated by C_α are replaced with the corresponding portions as annotated by C_β, thus simulating the presence of incrementally higher amounts of noise. For each setting we compute the intercoder agreement between the two training sets; this allows us to study the relative loss in extraction accuracy as a function of the agreement between authoritative and non-authoritative assessor as measured on the training set. Since in many practical situations it is easy to compute (or estimate) the intercoder disagreement between (a) the coder to whom we would ideally entrust the annotation task (e.g., a senior expert in the organization), and (b) the coder to whom we can indeed entrust it given time and cost constraints (e.g., a junior member of staff), this will give the reader a sense of how much intercoder disagreement generates how much loss in extraction accuracy.

The rest of the paper is organized as follows. Section 2 reviews related work on information extraction from clinical documents, and on establishing the relations between training data quality and extraction accuracy. In Sections 3 and 4 we describe experiments that attempt to quantify the degradation in extraction accuracy that derives from low-quality training data, with Section 3 devoted to spelling out the experimental setting and Section 4 devoted instead to presenting and discussing the results. Section 5 concludes, discussing avenues for further research.

[3]In the domain of classification the authoritative and non-authoritative settings have also been called *self-classification* and *cross-classification*, respectively [Webber and Pickens 2013]. We depart from this terminology in order to avoid any confusion with *self-learning* (which refers to retraining a classifier by using, as additional training examples, examples the classifier itself has classified) and *cross-lingual classification* (which denotes a variant of text classification which exploits synergies between training data expressed in different languages).

2. RELATED WORK

2.1. Information extraction from clinical documents

Most works on information extraction from clinical documents rely on methods based on supervised learning, i.e., methods for training an information extraction system from manually annotated examples. Support vector machines (SVMs – [Jiang et al. 2011; Li et al. 2008; Sibanda et al. 2006]), hidden Markov models (HMMs – [Li et al. 2010]), and (especially) conditional random fields (CRFs – [Esuli et al. 2013; Gupta et al. 2014; Jiang et al. 2011; Jonnalagadda et al. 2012; Li et al. 2008; Patrick and Li 2010; Torii et al. 2011; Wang and Patrick 2009]) have been the learners of choice in this field, due to their good performance and to the existence of publicly available implementations.

In recent years, research on the analysis of clinical texts has been further boosted by the existence of "shared tasks" on this topic, such as the seminal i2b2 series ("Informatics for Integrating Biology and the Bedside" – [Sun et al. 2013; Uzuner et al. 2012; Uzuner et al. 2011]), the 2013 [Suominen et al. 2013] and 2014 [Kelly et al. 2014] editions of ShARe/CLEF eHealth, and the Semeval-2014 Task 7 "Analysis of Clinical Text" [Pradhan et al. 2014]. In these shared tasks the goal is to competitively evaluate information extraction tools that recognise mentions of various concepts of interest (e.g., mentions of diseases and disorders) as appearing in discharge summaries, and in electrocardiogram reports, echocardiograph reports, and radiology reports.

2.2. Low-quality training data and prediction accuracy

The literature on the effects of suboptimal training data quality on prediction accuracy is extremely scarce, even within the machine learning literature at large. An early such study is [Rossin and Klein 1999], which looks at these issues in the context of learning to predict prices of mutual funds from economic indicators. Differently from us, the authors work with noise artificially inserted in the training set, and not with naturally occurring noise. From experiments run with a linear regression model they reach the bizarre conclusion that "the predictive accuracy (...) is better when errors exist in training data than when training data are free of errors", while the opposite conclusion is (somehow more expectedly) reached from experiments run with a neural networks model. A similar study, in which the context is predicting the average air temperature in distributed heating systems, was carried out in [Jassar et al. 2009]; yet another study, in which the goal was predicting the production levels of palm oil via a neural network, is [Khamis et al. 2005].

In the context of a biomedical information extraction task[4] Haddow and Alex [2008] examined the situation in which training data annotated by two different coders are available, and they found that higher accuracy is obtained by using both versions at the same time than by attempting to reconcile them or using just one of them. Their use case is different from ours, since in the case we discuss we assume that only one set of annotations, those of the non-authoritative coder, are available as training data. Note also that training data independently annotated by more than one coder are rarely available in practice.

Closer to our application context, Esuli and Sebastiani [2013] have thoroughly studied the effect of suboptimal training data quality in text classification. However, in their case the degradation in the quality of the training data is obtained, for mere experimental purposes, via the insertion of artificial noise, due to the fact that their datasets did not contain data annotated by more than one coder. As a result, it is

[4]Biomedical IE is different from clinical IE, in that the latter (unlike the former) is usually characterized by idiosyncratic abbreviations, ungrammatical sentences, and sloppy language in general. See [Meystre et al. 2008, p. 129] for a discussion of this point.

not clear how well the type of noise they introduce models naturally occurring noise. Webber and Pickens [2013] also address the text classification task (in the context of e-discovery from legal texts), but differently from [Esuli and Sebastiani 2013] they work with naturally occurring noise; differently from the present work, the multiply-coded training data they use were coded by one coder known to be an expert coder and another coder known to be a junior coder. Our work instead (a) focuses on information extraction, and (2) does not make any assumption on the relative level of expertise of the two coders.

3. EXPERIMENTAL SETTING

3.1. Basic notation and terminology

Let us fix some basic notation and terminology. Let $\mathbf{X}$ be a set of texts, where we view each text $\mathbf{x} \in \mathbf{X}$ as a sequence $\mathbf{x} = \langle x_1, \ldots, x_{|\mathbf{x}|} \rangle$ of *textual units* (or simply *t-units*), such that odd-numbered t-units are *tokens* (i.e., word occurrences) and even-numbered t-units are *separators* (i.e., sequences of blanks and punctuation symbols), and such that x_{t_1} occurs before x_{t_2} in the text (noted $x_{t_1} \preceq x_{t_2}$) if and only if $t_1 \leq t_2$. We dub $|\mathbf{x}|$ the *length* of the text. Let $C = \{c_1, \ldots, c_m\}$ be a predefined set of *concepts* (a.k.a. *tags*, or *markables*), or *tagset*. We take *information extraction* (IE) to be the task of determining, for each $\mathbf{x} \in \mathbf{X}$ and for each $c_r \in C$, a sequence $\mathbf{y}_r = \langle y_{r1}, \ldots, y_{r|\mathbf{x}|} \rangle$ of *labels* $y_{rt} \in \{c_r, \overline{c}_r\}$, which indicates which t-units in the text are labelled with tag c_r and which are not. Since each $c_r \in C$ is dealt with independently of the other concepts in C, we hereafter drop the r subscript and, without loss of generality, treat IE as the *binary* task of determining, given text $\mathbf{x}$ and concept c, a sequence $\mathbf{y} = \langle y_1, \ldots, y_{|\mathbf{x}|} \rangle$ of labels $y_t \in \{c, \overline{c}\}$.

T-units labelled with a concept c usually come in coherent sequences, or "mentions". Hereafter, a *mention* σ of text $\mathbf{x}$ for concept c will be a pair (x_{t_1}, x_{t_2}) consisting of a start token x_{t_1} and an end token x_{t_2} such that (i) $x_{t_1} \preceq x_{t_2}$, (ii) all t-units $x_{t_1} \preceq x_t \preceq x_{t_2}$ are labelled with concept c, and (iii) the token that immediately precedes x_{t_1} and the one that immediately follows x_{t_2} are *not* labelled with concept c. In general, a text $\mathbf{x}$ may contain zero, one, or several mentions for concept c.

In the above definitions we consider separators to be also the object of tagging in order for the IE system to correctly identify consecutive mentions. For instance, given the expression "Barack Obama, Hillary Clinton" the perfect IE system will attribute the PersonName tag to the tokens "Barack", "Obama", "Hillary", "Clinton", and to the separators (in this case: blank spaces) between "Barack" and "Obama" and between "Hillary" and "Clinton", but *not* to the separator ", " between "Obama" and "Hillary". If the IE system does so, this means that it has correctly identified the boundaries of the two mentions "Barack Obama" and "Hillary Clinton"[5].

3.2. Dataset

The dataset we have used to test the ideas discussed in the previous sections is the UmbertoI(RadRep) dataset first presented in [Esuli et al. 2013], consisting of a set of 500 free-text mammography reports written (in Italian) by medical personnel of the

[5]Note that the above notation is not able to represent "discontiguous mentions", i.e., mentions containing gaps, and "overlapping mentions", i.e., multiple mentions sharing one or more tokens. This is not a serious limitation for our research, since the above notation can be easily extended to deal with both phenomena (e.g., by introducing unique mention identifiers and having each t-unit be associated with zero, one, or several such identifiers), and since the dataset we use for our experimentation contains neither discontinuous nor overlapping mentions. We prefer to keep the notation simple, since the issue we focus on in this paper (the consequences on extraction accuracy of suboptimal training data quality) can be considered largely independent of the expressive power of the markup language.

Table I. The distribution of annotations across concepts, at token and mention level, for each coder.

	DEE	IES	ITE	ECH	LLO	TFU	DEP	BIR	PAE	Total
Tokens annotated by Coder1	4819	1529	7410	237	1811	1672	585	466	1723	18529
Tokens annotated by Coder2	7351	1723	7630	1329	2544	2670	1127	448	3495	24822
Mentions annotated by Coder1	204	140	190	51	164	149	19	128	344	1045
Mentions annotated by Coder2	282	145	188	102	193	171	26	103	399	1210

Istituto di Radiologia of Policlinico Umberto I, Roma, IT. The dataset is annotated according to 9 concepts relevant to the field of radiology and mammography: BIR ("Outcome of the BIRADS test"), ITE ("Technical Info"), IES ("Indications obtained from the Exam"), TFU ("Followup Therapies"), DEE ("Description of Enhancement"), PAE ("Presence/Absence of Enhancements"), ECH ("Outcomes of Surgery"), DEP ("Prosthesis Description"), and LLO ("Locoregional Lymph Nodes"). Note that we had no control on the design of the concept set, on its range, and on its granularity, since the choice of the concepts was entirely under the responsibility of Policlinico Umberto I. We thus take both the concept set and the dataset as given.

Mentions of these concepts are present in the reports according to fairly irregular patterns. In particular, a given concept (a) need not be instantiated in all reports, and (b) may be instantiated more than once (i.e., by more than one mention) in the same report. Mentions instantiating different concepts may overlap, and the order of presentation of the different concepts varies across the reports. On average, there are 0.87 mentions for each concept in a given report, and the average mention length is 17.33 words.

The reports were annotated by two equally expert radiologists, Coder1 and Coder2; 191 reports were annotated by Coder1 only, 190 reports were annotated by Coder2 only, and 119 reports were annotated independently by Coder1 and Coder2. From now on we will call these sets `1-only`, `2-only` and `Both`, respectively; `Both(1)` will identify the `Both` set as annotated by Coder1, and `Both(2)` will identify the `Both` set as annotated by Coder2. The annotation activity was preceded by an alignment phase, in which Coder1 and Coder2 jointly annotated 20 reports (not included in this dataset) in order to align their understanding of the meaning of the concepts.

Table I reports the distribution of annotations across concepts, at token and mention level, for the two coders; see [Esuli et al. 2013, Section 4.2] for a more detailed description of the UmbertoI(RadRep) dataset that includes additional stats[6].

3.3. Learning algorithms

As the learning algorithms we have tested both *linear-chain conditional random fields* (LC-CRFs - [Lafferty et al. 2001; Sutton and McCallum 2007; Sutton and McCallum 2012]), in Charles Sutton's GRMM implementation[7], and *hidden Markov support vector machines* (HM-SVMs - [Altun et al. 2003]), in Thorsten Joachims's SVM^{hmm} implementation[8]. Both are supervised learning algorithms explicitly devised for *sequence labelling*, i.e., for learning to label (i.e., to annotate) items that naturally occur in sequences and such that the label of an item may depend on the features and/or on the labels of other items that precede or follow it in the sequence (which is indeed the case for the tokens in a text)[9]. LC-CRFs are members of the class

[6]No other dataset is used in this paper since we were not able to locate another dataset of annotated clinical texts that contains reports annotated by more than one coder and is at the same time publicly available.
[7]http://mallet.cs.umass.edu/grmm/
[8]http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html
[9]Note that only tokens, and not separators, are explicitly labelled. The reason is that both LC-CRFs and HM-SVMs actually use the so-called IOB labelling scheme, according to which, for each concept $c_r \in C$, a

of *graphical models*, a family of probability distributions that factorize according to an underlying graph [Wainwright and Jordan 2008]; see [Sutton and McCallum 2012] for a full mathematical explanation of LC-CRFs. HM-SVMs are an instantiation of "SVMs for structured output prediction" (SVM^{struct}) [Tsochantaridis et al. 2005] for the sequence labelling task, and have already been used in clinical information extraction (see e.g., [Tang et al. 2012; Zhang et al. 2014]). In HM-SVMs the learning procedure is based on a large-margin approach typical of SVMs, which, differently from LC-CRFs, can learn non-linear discriminant functions via kernel functions.

Both learners need each token x_t to be represented by a vector $\mathbf{x}_t$ of features[10]. In this work we have used a set of features which includes one feature representing the word of which the token is an instance, one feature representing its stem, one feature representing its part of speech, eight features representing its prefixes and suffixes (the first and the last n characters of the token, with $n = 1, 2, 3, 4$), one feature representing information on token capitalization (i.e., whether the token is all uppercase, all lowercase, first letter uppercase, or mixed case), and 4 "positional" features [Esuli et al. 2013, Section 3.3] that indicate in which half, 3rd, 4th, or 5th, respectively, of the text the token occurs in.

3.4. Evaluation measures

3.4.1. Classification accuracy. As a measure of classification accuracy we use, similarly to [Esuli et al. 2013], the token-and-separator variant (proposed in [Esuli and Sebastiani 2010]) of the well-known F_1 measure, according to which an information extraction system is evaluated on an event space consisting of all the t-units in the text. In other words, each t-unit x_t (rather than each *mention*, as in the traditional "segmentation F-score" model [Suzuki et al. 2006]) counts as a true positive, true negative, false positive, or false negative for a given concept c_r, depending on whether x_t belongs to c_r or not in the predicted annotation and in the true annotation. This model has the advantage that it credits a system for partial success (i.e., degree of overlap between a predicted mention and a true mention for the same concept), and that it penalizes both overannotation and underannotation.

As is well-known, F_1 is the harmonic mean of *precision* ($\pi = \frac{TP}{TP+FN}$) and *recall* ($\rho = \frac{TP}{TP+FP}$), and is defined as

$$F_1 = \frac{2\pi\rho}{\pi+\rho} = \frac{2 \cdot \frac{TP}{TP+FN} \cdot \frac{TP}{TP+FP}}{\frac{TP}{TP+FN} + \frac{TP}{TP+FP}} = \frac{2TP}{2TP+FP+FN} \quad (1)$$

where TP, FP, and FN stand for the numbers of true positives, false positives, and false negatives, respectively. It is easy to observe that F_1 is equivalent to TP divided by the arithmetic mean of the actual positives and the predicted positives (or, alternatively, the product of π and ρ divided by their arithmetic mean). Note that F_1 is undefined when $TP = FP = FN = 0$; in this case we take F_1 to equal 1, since the system has correctly annotated all t-units as negative.

token can be labelled as B_r (the beginning token of a mention of c_r), I_r (a token which is inside a mention of c_r but is not its beginning token), and O_r (a token that is outside any mention of c_r). As a result, a separator is (implicitly) labelled with concept c_r if and only if it precedes a token labelled with I_r. We may think of the notation of Section 3.1 as an abstract markup language, and of the IOB notation as a concrete markup language, in the sense that the notation of Section 3.1 is easier to understand (and will also make the evaluation measure discussed in Section 3.4.1 easier to understand) while IOB is actually used by the learning algorithms. The two notations are equivalent in expressive power.

[10]Note that only tokens, and not separators, are explicitly represented in vectorial form, the reasons being the same already discussed in Footnote 9.

We compute F_1 across the entire test set, i.e., we generate a single contingency table by putting together all t-units in the test set, irrespectively of the document they belong to. We then compute both *microaveraged* F_1 (denoted by F_1^μ) and *macroaveraged* F_1 (F_1^M). F_1^μ is obtained by (i) computing the concept-specific values TP_r, FP_r and FN_r, (ii) obtaining TP as the sum of the TP_r's (same for FP and FN), and then (iii) applying Equation 1. F_1^M is obtained by first computing the concept-specific F_1 values and then averaging them across the c_r's.

3.4.2. Intercoder agreement. *Intercoder agreement* (ICA), or the lack thereof (*intercoder disagreement*), has been widely studied for over a century (see e.g., [Krippendorff 2004] for an introduction). As a phenomenon, disagreement among coders naturally occurs when units of content need to be annotated by humans according to their semantics (i.e., when the occurrences of specific concepts need to be recognized within these units of content). Such disagreement derives from the fact that semantic content is a highly subjective notion: different coders might disagree with each other as to what the semantics of, say, a given piece of text is, and it is even the case that the same coder might at times disagree with herself (i.e., return different codes when coding the same unit of content at different times).

ICA may be measured by the relative frequency of the units of content on which coders agree, usually normalized by the probability of chance agreement. Many metrics for ICA have been proposed over the years, "Cohen's kappa" probably being the most famous and widely used ("Scott's pi" and "Krippendorff's alpha" are others); sometimes (see e.g., [Chapman and Dowling 2006; Esuli et al. 2013]) functions that were not explicitly developed for measuring ICA (such as F_1, that was developed for measuring binary classification accuracy) are used. The levels of ICA that are recorded in actual experiments vary a lot across experiments, types of content, and types of concepts that are to be recognized in the units of content under investigation. This extreme variance depends on factors such as "annotation domain, number of categories in a coding scheme, number of annotators in a project, whether annotators received training, the intensity of annotator training, the annotation purpose, and the method used for the calculation of percentage agreements" [Bayerl and Paul 2011]. The actual meaning of the concepts the coders are asked to recognize is a factor of special importance, to the extent that a concept on which very low levels of ICA are reached may be deemed, because of this very fact, ill-defined.

For measuring intercoder agreement we use Cohen's kappa (noted κ), defined as

$$\begin{aligned}
\kappa &= \frac{P(A) - P(E)}{1 - P(E)} \\
&= \frac{(P(p = t = c) + P(p = t = \overline{c})) - (P(p = c)P(t = c) + P(p = \overline{c})P(t = \overline{c}))}{1 - (P(p = c)P(t = c) + P(p = \overline{c})P(t = \overline{c}))} \\
&= \frac{\frac{TP+TN}{n} - ((\frac{TP+FP}{n})(\frac{TP+FN}{n}) + (\frac{FN+TN}{n})(\frac{FP+TN}{n}))}{1 - ((\frac{TP+FP}{n})(\frac{TP+FN}{n}) + (\frac{FN+TN}{n})(\frac{FP+TN}{n}))}
\end{aligned} \tag{2}$$

where $P(A)$ denotes the probability (i.e., relative frequency) of agreement, $P(E)$ denotes the probability of chance agreement, and n is the total number of examples (see [Artstein and Poesio 2008; Di Eugenio and Glass 2004] for details); here, we use the shorthand $p = c$ (resp., $t = c$) to mean that the predicted label (resp., true label) is c (analogously for $\overline{c}$). We opt for kappa since it is the most widely known, and best understood, measure of ICA. For Cohen's kappa too we work at the t-unit level, i.e., for

each t-unit x_t we record whether the two coders agree on whether x_t is labelled or not with the concept c of interest.

Incidentally, note that (as observed in [Esuli and Sebastiani 2010]) we can compute Cohen's kappa only thanks to the fact that (as discussed in Section 3.4.1) we conduct our evaluation at the t-unit level (rather at the mention level). Those who conduct their evaluation at the mention level (e.g., [Chapman and Dowling 2006]) find that they are unable to do so, since in order to be defined kappa needs the notion of a true negative to be also defined, and this is undefined at the mention level. Evaluation at the mention level thus prevents the use of kappa and leaves F_1 as the only choice.

4. RESULTS

4.1. Experimental protocol

In [Esuli et al. 2013], experiments on the UmbertoI(RadRep) dataset were run using either `1-only` and/or `2-only` (i.e., the portions of the data that only one coder had annotated) as training data and `Both(1)` and/or `Both(2)` (i.e., the portion of the data that both coders had annotated, in both versions) as test data.

In this paper we switch the roles of training set and test set, i.e., use `Both(1)` or `Both(2)` as training set (since for the purpose of this paper we need *training* data with multiple, alternative annotations) and `1-only` or `2-only` as test set. Specifically, we run two batches of experiments, Batch1 and Batch2. In Batch1 Coder1 plays the role of the authoritative coder (C_α) and Coder2 plays the role of the non-authoritative coder (C_β), while in Batch2 Coder2 plays the role of C_α and Coder1 plays the role of C_β.

Each of the two batches of experiments is composed of:

(1) An experiment using the authoritative setting, i.e., both training and test data are annotated by C_α. This means training on `Both(1)` and testing on `1-only` (Batch1) and training on `Both(2)` and testing on `2-only` (Batch2).
(2) An experiment using the non-authoritative setting, i.e., training data annotated by C_β and test data annotated by C_α. This means training on `Both(2)` and testing on `1-only` (Batch1) and training on `Both(1)` and testing on `2-only` (Batch2).
(3) Experiments using the partially authoritative setting, i.e., test data annotated by C_α, and training data annotated in part by C_β ($\lambda\%$ of the training documents, chosen at random) and in part by C_α (the remaining $(100 - \lambda)\%$ of the training documents). We call λ the *corruption ratio* of the training set; $\lambda = 0$ obviously corresponds to the fully authoritative setting while $\lambda = 100$ corresponds to the non-authoritative setting.
We run experiments for each $\lambda \in \{10, 20, ..., 80, 90\}$ by monotonically adding, for increasing values of λ, new randomly chosen elements (10% at a time) to the set of training documents annotated by C_β. Since the choice of training data annotated by C_β is random, we repeat the experiment 10 times for each value of $\lambda \in \{10, 20, ..., 80, 90\}$, each time with a different random such choice.

For each of the above train-and test experiment we compute the intercoder agreement $\kappa(Tr, corr_\lambda(Tr))$ between the non-corrupted version of the training set Tr and the (partially or fully) corrupted version $corr_\lambda(Tr)$ for a given value of λ. We then take the average among the 10 values of $\kappa(Tr, corr_\lambda(Tr))$ deriving from the 10 different experiments run for a given value of λ and denote it as $\kappa(\lambda)$; this value indicates the average intercoder agreement that derives by "corrupting" $\lambda\%$ of the documents in the training set, i.e., by using for them the annotations performed by the non-authoritative coder.

For each of the above train-and test experiment we also compute the extraction accuracy (via both F_1^μ and F_1^M) and the relative loss in extraction accuracy that results from the given corruption ratio.

Table II. Extraction accuracy for the authoritative setting ($\lambda = 0$) and non-authoritative setting ($\lambda = 100$), for the LC-CRFs and HM-SVMs learners, for both batches of experiments (and for the average across the two batches), along with the resulting intercoder agreement values expressed as $\kappa(\lambda)$. Percentages indicate the loss in extraction accuracy resulting from moving from $\lambda = 0$ to $\lambda = 100$.

			LC-CRFs				HM-SVMs			
	λ	$\kappa(\lambda)$	F_1^μ		F_1^M		F_1^μ		F_1^M	
Batch1	0	1.000	0.783		0.674		0.820		0.693	
	100	0.742	0.765	(-2.35%)	0.668	(-0.90%)	0.786	(-4.33%)	0.688	(-0.73%)
Batch2	0	1.000	0.808		0.752		0.817		0.754	
	100	0.742	0.733	(-10.23%)	0.654	(-14.98%)	0.733	(-11.46%)	0.625	(-20.64%)
Average	0	1.000	0.795		0.713		0.819		0.724	
	100	0.742	0.749	(-6.14%)	0.661	(-7.87%)	0.760	(-7.76%)	0.657	(-10.20%)

4.2. Results and discussion

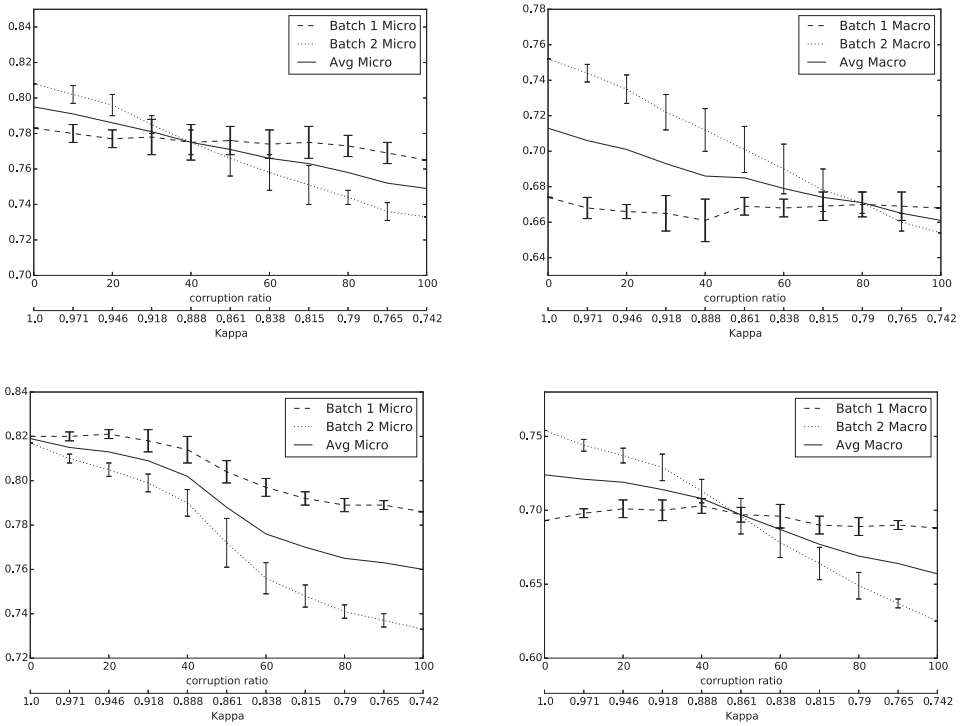
Table II reports extraction accuracy figures for the authoritative and non-authoritative settings, for both learners, both batches of experiments, and along with the resulting intercoder agreement values. Figure 1 illustrates the results of our experiments by plotting F_1 as a function of the corruption ratio λ, using LC-CRFs and HM-SVMs as the learning algorithm, respectively; for each value of λ, the corresponding level of interannotator agreement $\kappa(\lambda)$ (as averaged across the two batches) is also indicated. Figure 2 plots instead precision and recall as a function of λ for the LC-CRFs experiments, while Figure 3 does the same for the HM-SVMs experiments.

4.2.1. Macroaveraged values are lower than microaveraged ones. A first fact to be observed is that macroaveraged (F_1^M) results are always lower than the corresponding microaveraged (F_1^μ) results. This is unsurprising, and conforms to a well-known pattern. In fact, microaveraged effectiveness scores are heavily influenced by the accuracy obtained on the concepts most frequent in the test set (i.e., on the ones that label many test t-units); for these concepts accuracy tends to be higher, since these concepts also tend to be more frequent in the training set. Conversely, in macroaveraged effectiveness measures, each concept counts the same, which means that the low-frequency concepts (which tend to be the low-performing ones too) have as much of an impact as the high-frequency ones. See [Debole and Sebastiani 2005, pp. 591–593] for a thorough discussion of this point in a text classification context.

4.2.2. HM-SVMs outperform LC-CRFs. A second fact that emerges is that HM-SVMs outperform LC-CRFs, on both batches, both settings (authoritative and non-authoritative), and both evaluation measures (F_1^μ and F_1^M); e.g., on the authoritative setting, and as an average across the two batches, HM-SVMs obtain $F_1^\mu = 0.819$ (while LC-CRFs obtain 0.795) and $F_1^M = 0.724$ (while LC-CRFs obtain 0.713). Aside from their different levels of effectiveness, the two learners behave in a qualitatively similar way as a function of λ, as evident from a comparison of Figures 2 and 3. However, we will not dwell on this fact any further since the relative performance of the learning algorithms is not the main focus of the present study; as will be evident in the discussion that follows, most insights obtained from the LC-CRFs experiments are qualitatively confirmed by the HM-SVMs experiments, and vice versa.

4.2.3. Coder1 generates less accuracy than Coder2. A third fact that may be noted (from Table II) is that, for $\lambda = 0$, there is a substantive difference in accuracy values between the two coders, with Coder2 usually generating higher accuracy than Coder1. This fact can be especially appreciated at the macroaveraged level (where for LC-CRFs we have $F_1^M = 0.674$ for Coder1 and $F_1^M = 0.752$ for Coder2, and for HM-SVMs we

Fig. 1. Microaveraged F_1 (left) and macroaveraged F_1 (right) as a function of the fraction λ of the training set that is annotated by C_β instead of C_α ("corruption ratio"), using **LC-CRFs** (top) and **HM-SVMs** (bottom) as learning algorithms. The dashed line represents the experiments in Batch1, the dotted line represents those in Batch2, and the solid one represents the average between the two batches. The vertical bars indicate, for each $\lambda \in \{10, 20, ..., 80, 90\}$, the standard deviation across the 10 runs deriving from the 10 random choices of $corr_\lambda(Tr)$.



have $F_1^M = 0.693$ for Coder1 and $F_1^M = 0.754$ for Coder2), while the difference is less clearcut at the microaveraged level (where for LC-CRFs we have $F_1^M = .0.783$ for Coder1 and $F_1^M = 0.808$ for Coder2, and for HM-SVMs we have $F_1^M = 0.820$ for Coder1 and $F_1^M = 0.817$ for Coder2); this indicates that the codes where Coder2 especially shines are the low-frequency ones.

In principle, there might be several reasons for this difference in accuracy values between the two coders. The documents in `2-only` might be "easier" to code automatically than those in `1-only`; or the distributions of `Both(1)` and `1-only` might be less similar to each other than the distributions of `Both(2)` and `2-only`, thus verifying the i.i.d. assumption to a higher degree; or Coder2 might simply be more self-consistent in her annotation style than Coder1.

In order to check whether the last of these three hypotheses is true we have performed four k-fold cross-validation (k-FCV) experiments (for `Both(1)` and `Both(2)`, and for LC-CRFs and HM-SVMs, in all combinations), using $k = 20$. Intuitively, a higher accuracy value resulting from a k-FCV test means a higher level of self-consistency, since if the same coding style is consistently used to label a dataset, a system tends to encounter in the testing phase the same labelling patterns it has encountered in the training phase, which is conducive to higher accuracy. Of course, the results of such a

Fig. 2. Microaveraged (left) and macroaveraged (right) precision (top) and recall (bottom) as a function of the fraction λ of the training set that is annotated by C_β instead of C_α ("corruption ratio"), using **LC-CRFs** as a learning algorithm.

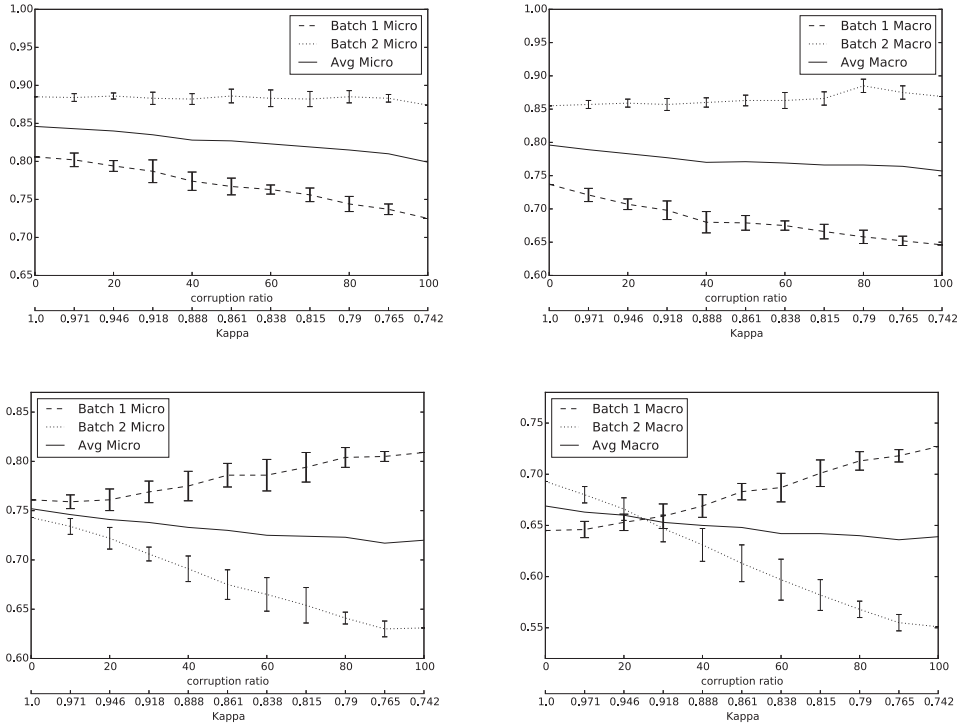


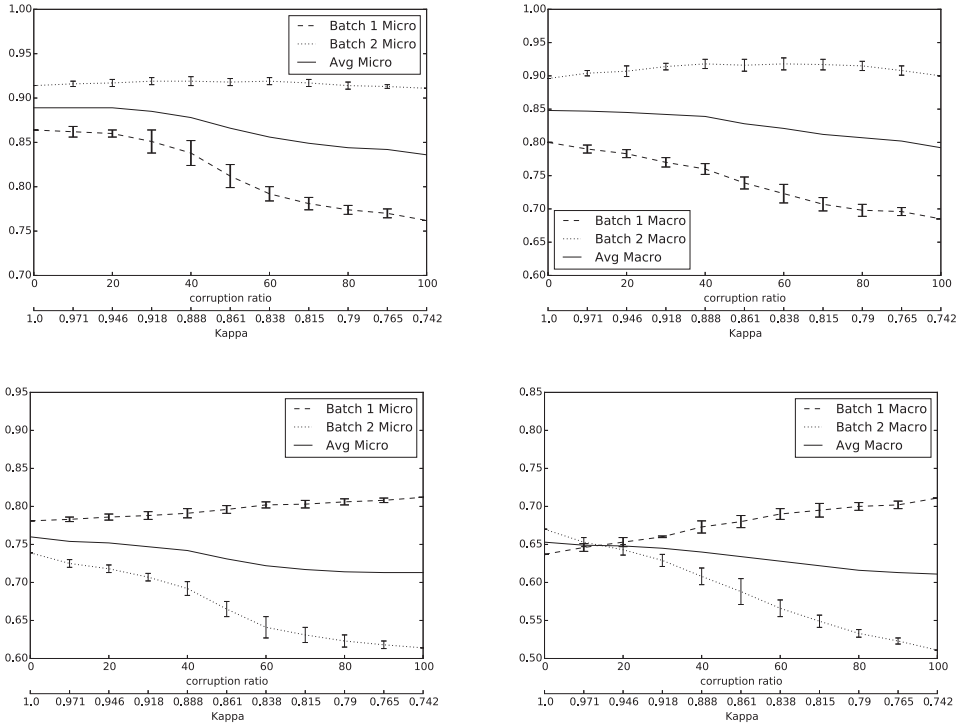
Table III. Results of the 20-fold cross-validation tests on `Both(1)` and `Both(2)`, for LC-CRFs and HM-SVMs.

	LC-CRFs		HM-SVMs	
	F_1^μ	F_1^M	F_1^μ	F_1^M
`Both(1)`	0.829	0.735	0.842	0.737
`Both(2)`	0.838	0.771	0.850	0.787

test are difficult to interpret if the goal is to assess the self-consistency of a coder in *absolute* terms (since we do not know what values of F_1 correspond to what levels of self-consistency), but they are not if the goal is simply to establish which of the two is the more self-consistent, since the two experiments are run on the same documents. The results of our two k-FCV experiments are reported in Table III. From this table we can see that the accuracy on `Both(2)` is substantially higher than the one obtained on `Both(1)`, thus indicating that Coder2 is indeed more self-consistent than Coder1. This is thus the likely explanation of the higher levels of accuracy obtained on the dataset annotated, for both training and test, by Coder2.

4.2.4. Overannotation and underannotation. A fourth, even more interesting fact we may observe is that accuracy as a function of the corruption ratio varies much less for Batch1 than for Batch2, since for this latter we witness a much more substantial drop in going

Fig. 3. Microaveraged (left) and macroaveraged (right) precision (top) and recall (bottom) as a function of the fraction λ of the training set that is annotated by C_β instead of C_α ("corruption ratio"), using **HM-SVMs** as a learning algorithm.



from $\lambda = 0$ to $\lambda = 100$. We conjecture that this may be due to the different annotation style of the two coders; the rest of this subsection will be devoted to explaining the rationale of this conjecture.

As evident from Table I, Coder2 annotates, as instances of the concepts of interest, more mentions (+15.7%) and also more tokens per mention (+15.6%) than Coder1; relatively to each other, Coder1 is thus an *underannotator* while Coder2 is an *overannotator*. Since, as noted in Section 1, learning algorithms learn to replicate the subjective annotation style of their supervisors, a system trained on data annotated by an overannotator will itself tend to overannotate; conversely, a system trained by an underannotator will itself tend to underannotate.

Overannotation results in more true positives and more false positives. The plots in Figures 2 and 3 show that when, as a consequence of increased values of λ, the number of training documents annotated by an overannotator increases (as is the case of Batch1), precision suffers somehow (due to the fact that, along with more true positives, there are also more false positives), but this is compensated by an increase in recall (due to an increased number of true positives); as a result, as shown in Figure 1 (and in Table II too), the drop in F_1 resulting from moving to $\lambda = 0$ to $\lambda = 100$ is very limited. Figures 2 and 3 instead show that when, as a consequence of increased values of λ, the number of training documents annotated by an underannotator increases (as is the case for Batch2), recall drops substantially (due to the decreased number of true positives), and this drop is not compensated by the stability of precision (which is due

Table IV. Results of the approximate randomization test, measuring the statistical significance of the difference between the accuracy of the system trained at $\lambda = 0$ and the accuracy of the system trained at $\lambda = 100$. Results are reported for both learners (LC-CRFs and HM-SVMs), both batches, and both evaluation measures (F_1^μ and F_1^M).

	LC-CRFs		HM-SVMs	
	F_1^μ	F_1^M	F_1^μ	F_1^M
Batch1	0.0859	0.6207	0.0001	0.5040
Batch2	0.0001	0.0001	0.0001	0.0001

to the combined effect of a decrease in true positives and a decrease in false positives); as a result, as shown in Figure 1 (see also Table II), the drop in F_1 resulting from moving to $\lambda = 0$ to $\lambda = 100$ is much more substantial than for Batch1.

In order to check whether the decreases in accuracy between the $\lambda = 0$ and the $\lambda = 100$ settings is statistically significant we have performed an *approximate randomization test* (ART) [Chinchor et al. 1993]. In this test the difference is considered statistically significant if the resulting p value is < 0.05. Two advantages of the ART are that

(1) unlike the t-test, the ART does not require the data to be normally distributed;
(2) unlike the Wilcoxon signed-rank test, the ART can be applied to multivariate non-linear evaluation measures such as F_1 [Yeh 2000].

The results of our statistical significance tests are reported in Table IV. These results essentially confirm the observations above, i.e., that *in Batch2 the drop in performance resulting from having the training set annotated by the non-authoritative coder (instead of the authoritative one) is not statistically significant*, while (with the exception of the F_1^μ results for HM-SVMs) it is statistically significant for Batch1.

4.2.5. Caveats. The experiments discussed in this paper do not allow us to reach hard conclusions about the robustness of information extraction systems to imperfect training data quality, for several reasons:

(1) The results obtained should be confirmed by additional experiments carried out on other datasets; unfortunately, as noted in Footnote 6, we were not able to locate any other publicly available dataset with the required characteristics (that is, containing at least some doubly annotated documents).
(2) The dataset used here is representative of only a specific type of imperfect training data quality, i.e., the one deriving from the fact that the training data were annotated by a coder different (albeit equally expert) from the one who annotated the test set. Other types do exist, however, as noted in the introduction.
(3) Even the results reported here are somehow contradictory, since a statistically significant drop in performance was observed in Batch1 while no such statistically significant drop was observed in Batch2.

However, one interesting fact that has emerged from the present study (and that will need to be confirmed by additional experiments, should other datasets become available) is that, as argued in detail in Section 4.2.4, the lack of a statistically significant drop in performance observed in Batch2 seems to be due to the fact that the non-authoritative coder who annotated the training set had an *overannotating* behaviour. This *might* suggest (emphasis meaning that prudence should be exercised) that, should there be a need for having a training set annotated by someone different from the au-

thoritative coder, underannotation should be discouraged much more than overannotation.

5. CONCLUSIONS

Few researchers have investigated the loss in accuracy that occurs when a supervised learning algorithm is fed with training data of suboptimal quality. We have done this for the first time in the case of information extraction systems (trained via supervised learning) as applied to the detection of mentions of concepts of interest in medical notes. Specifically, we have tested to what extent extraction accuracy suffers when the person who has annotated the test data (the "authoritative coder"), whom we must assume to be the person to whose judgment we conform irrespectively of her level of expertise, is different from the person who has labelled the training data (the "non-authoritative coder"). Our experimental results, that we have obtained on a dataset of 500 mammography reports annotated according to 9 concepts of interest, are somehow surprising, since they indicate that the resulting drop in accuracy is not always statistically significant. In our experiments, no statistically significant drop was observed when the non-authoritative coder had a tendency to overannotate, while a substantial, statistically significant drop was observed when the non-authoritative coder was an underannotator; however, experiments on more doubly (or even multiply) annotated datasets will be needed to confirm or disconfirm these initial findings. Since labelling cost is an important issue in the generation of training data (with senior coders costing much more than junior ones, and with internal coders costing much more than "mechanical turkers"), results of this kind may give important indications as to the cost-effectiveness of low-cost annotation work.

This paper is a first attempt to investigate the impact of less-than-sterling training data quality on the accuracy of medical concept extraction systems, and more work is needed to validate the conjectures that we have made based on our experimental results. As repeatedly mentioned in this paper, one limit of the present work is the fact that only one dataset was used for the experiments. This was due to the unfortunate lack of other publicly available medical datasets that contain (at least a subset of) textual records independently labelled by two different coders; datasets with these characteristics have been used in the past in published research but are not made available to the rest of the scientific community. We hope that the increasing importance of text mining applications in clinical practice, and the importance of shared datasets for fostering advances in this field, will generate a new kind of awareness on the need to make more datasets available to the scientific community.

REFERENCES

Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Hidden Markov Support Vector Machines. In *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*. Washington DC, USA, 3–10.

Ron Artstein and Massimo Poesio. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics* 34, 4 (2008), 555–596.

Petra S. Bayerl and Karsten I. Paul. 2011. What Determines Inter-Coder Agreement in Manual Annotations? A Meta-Analytic Investigation. *Computational Linguistics* 37, 4 (2011), 699–725.

Wendy W. Chapman and John N. Dowling. 2006. Inductive creation of an annotation schema for manually indexing clinical conditions from emergency department reports. *Journal of Biomedical Informatics* 39, 2 (2006), 196–208.

Nancy Chinchor, David D. Lewis, and Lynette Hirschman. 1993. Evaluating message understanding systems: An analysis of the third message understanding conference (MUC-3). *Computational Linguistics* 19, 3 (1993), 409–449.

Franca Debole and Fabrizio Sebastiani. 2005. An Analysis of the Relative Hardness of Reuters-21578 Subsets. *Journal of the American Society for Information Science and Technology* 56, 6 (2005), 584–596.

Barbara Di Eugenio and Michael Glass. 2004. The kappa statistic: A second look. *Computational Linguistics* 30, 1 (2004), 95–101.

Andrea Esuli, Diego Marcheggiani, and Fabrizio Sebastiani. 2013. An Enhanced CRFs-based System for Information Extraction from Radiology Reports. *Journal of Biomedical Informatics* 46, 3 (2013), 425–435.

Andrea Esuli and Fabrizio Sebastiani. 2010. Evaluating Information Extraction. In *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2010)*. Padova, IT, 100–111.

Andrea Esuli and Fabrizio Sebastiani. 2013. Training Data Cleaning for Text Classification. *ACM Transactions on Information Systems* 31, 4 (2013).

Catherine Grady and Matthew Lease. 2010. Crowdsourcing document relevance assessment with Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Los Angeles, US, 172–179.

Sonal Gupta, Diana L. MacLean, Jeffrey Heer, and Christopher D Manning. 2014. Induced lexico-syntactic patterns improve information extraction from online medical forums. *Journal of the American Medical Informatics Association* 21, 5 (2014), 902–909.

Barry Haddow and Beatrice Alex. 2008. Exploiting Multiply Annotated Corpora in Biomedical Information Extraction Tasks. In *Proceedings of the 6th Conference on Language Resources and Evaluation (LREC 2008)*. Marrakech, MA.

Surinder Jassar, Zaiyi Liao, and Lian Zhao. 2009. Impact of Data Quality on Predictive Accuracy of ANFIS-based Soft Sensor Models. In *Proceedings of the 2009 IEEE World Congress on Engineering and Computer Science (WCECS 2009)*, Vol. II. San Francisco, US.

Min Jiang, Yukun Chen, Mei Liu, S. Trent Rosenbloom, Subramani Mani, Joshua C. Denny, and Hua Xu. 2011. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *Journal of the American Medical Informatics Association* 18, 5 (2011), 601–606.

Siddhartha Jonnalagadda, Trevor Cohen, Stephen Wu, and Graciela Gonzalez. 2012. Enhancing clinical concept extraction with distributional semantics. *Journal of Biomedical Informatics* 45, 1 (2012), 129–140.

Liadh Kelly, Lorraine Goeuriot, Hanna Suominen, Tobias Schreck, Gondy Leroy, Danielle L. Mowery, Sumithra Velupillai, Wendy W. Chapman, David Martínez, Guido Zuccon, and João R. M. Palotti. 2014. Overview of the ShARe/CLEF eHealth Evaluation Lab 2014. In *Proceedings of the 5th International Conference of the CLEF Initiative (CLEF 2014)*. Sheffield, UK, 172–191.

Azme Khamis, Zuhaimy Ismail, Khalid Haron, and Ahmad T. Mohammed. 2005. The effects of outliers data on neural network performance. *Journal of Applied Sciences* 5, 8 (2005), 1394–1398.

Klaus Krippendorff. 2004. *Content analysis: An introduction to its methodology*. Sage, Thousand Oaks, US.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning (ICML 2001)*. Williamstown, US, 282–289.

Dingcheng Li, Karin Kipper-Schuler, and Guergana Savova. 2008. Conditional Random Fields and Support Vector Machines for Disorder Named Entity Recognition in Clinical Texts. In *Proceedings of the ACL Workshop on Current Trends in Biomedical Natural Language Processing (BioNLP 2008)*. Columbus, US, 94–95.

Ying Li, Sharon Lipsky Gorman, and Noémie Elhadad. 2010. Section classification in clinical notes using supervised hidden Markov model. In *Proceedings of the 2nd ACM International Health Informatics Symposium*. Arlington, US, 744–750.

Stephane M. Meystre, Guerguana K. Savova, Karin C. Kipper-Schuler, and John F. Hurdle. 2008. Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research. In *IMIA Yearbook of Medical Informatics*, A. Geissbuhler and C. Kulikowski (Eds.). Schattauer Publishers, Stuttgart, DE, 128–144.

Jon Patrick and Ming Li. 2010. High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge. *Journal of the American Medical Informatics Association* 17 (2010), 524–527.

Sameer Pradhan, Noémie Elhadad, Wendy Chapman, Suresh Manandhar, and Guergana Savova. 2014. SemEval-2014 Task 7: Analysis of Clinical Text. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, IE, 54–62.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence (Eds.). 2009. *Dataset shift in machine learning*. The MIT Press, Cambridge, US.

Donald F. Rossin and Barbara D. Klein. 1999. Data Errors in Neural Network and Linear Regression Models: An Experimental Comparison. *Data Quality Journal* 5, 1 (1999).

Claude Sammut and Michael Harries. 2011. Concept drift. In *Encyclopedia of Machine Learning*, Claude Sammut and Geoffrey I. Webb (Eds.). Springer, Heidelberg, DE, 202–205.

Tawanda Sibanda, Tian He, Peter Szolovits, and Özlem Uzuner. 2006. Syntactically-informed semantic category recognition in discharge summaries. In *Proceedings of the Annual Symposium of the American Medical Informatics Association (AMIA 2006)*. Washington, US, 714–718.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and Fast - But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*. Honolulu, US, 254–263.

Weiyi Sun, Anna Rumshisky, and Özlem Uzuner. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. *Journal of the American Medical Informatics Association* 20, 5 (2013), 806–813.

Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W. Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R. South, Danielle L. Mowery, Gareth J. Jones, Johannes Leveling, Liadh Kelly, Lorraine Goeuriot, David Martinez, and Guido Zuccon. 2013. Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. In *Proceedings of the 4th International Conference of the CLEF Initiative (CLEF 2013)*. Valencia, ES, 212–231.

Charles Sutton and Andrew McCallum. 2007. An Introduction to Conditional Random Fields for Relational Learning. In *Introduction to Statistical Relational Learning*, Lise Getoor and Ben Taskar (Eds.). The MIT Press, Cambridge, US, 93–127.

Charles Sutton and Andrew McCallum. 2012. An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning* 4, 4 (2012), 267–373.

Jun Suzuki, Erik McDermott, and Hideki Isozaki. 2006. Training Conditional Random Fields with Multivariate Evaluation Measures. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL (ACL/COLING 2006)*. Sydney, AU, 217–224.

Buzhou Tang, Hongxin Cao, Yonghui Wu, Min Jiang, and Hua Xu. 2012. Clinical entity recognition using structural support vector machines with rich features. In *Proceedings of the 6th ACM International Workshop on Data and Text Mining in Biomedical Informatics (DTMBIO 2012)*. Maui, US, 13–20.

Manabu Torii, Kavishwar Wagholikar, and Hongfang Liu. 2011. Using machine learning for concept extraction on clinical documents from multiple data sources. *Journal of the American Medical Informatics Association* 18, 5 (2011), 580–587.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research* 6 (2005), 1453–1484.

Özlem Uzuner, Andreea Bodnari, Shuying Shen, Tyler Forbush, John Pestian, and Brett R. South. 2012. Evaluating the state of the art in coreference resolution for electronic medical records. *Journal of the American Medical Informatics Association* 19, 5 (2012), 786–791.

Özlem Uzuner, Brett R. South, Shuying Shen, and Scott L. DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association* 18, 5 (2011), 552–556.

Martin J. Wainwright and Michael I. Jordan. 2008. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning* 1, 1/2 (2008), 1–305.

Yefeng Wang and Jon Patrick. 2009. Cascading classifiers for named entity recognition in clinical notes. In *Proceedings of the RANLP 2009 Workshop on Biomedical Information Extraction*. Borovets, BG, 42–49.

William Webber and Jeremy Pickens. 2013. Assessor disagreement and text classifier accuracy. In *Proceedings of the 36th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2013)*. Dublin, IE, 929–932.

Alexander S. Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*. Saarbrücken, DE, 947–953.

Yaoyun Zhang, Jingqi Wang, Buzhou Tang, Yonghui Wu, Min Jiang, Yukun Chen, and Hua Xu. 2014. UTH_CCB: A report for SemEval 2014 – Task 7 Analysis of Clinical Text. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, IE, 802–806.