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

DIEGO MARCHEGGIANI, University of Amsterdam
FABRIZIO SEBASTIANI, Italian National Council of Research

In the last five years there has been a flurry of work on information extraction from clinical documents, that is, 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. Many of these research works are about methods based on supervised learning, that is, 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 for the clinical domain. 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 article, 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, equally expert in the subject matter. 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. While the dataset used in the present work originated in a clinical context, the issues we study in this work are of more general interest.

Categories and Subject Descriptors: H.3.3 Information systems [**Information retrieval**]: Retrieval tasks and goals—*Clustering and Classification*; I.2.6 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. 2017. On the Effects of Low-Quality Training Data on Information Extraction from Clinical Reports. *J. Data and Information Quality* 9, 1, Article 1 (September 2017), 25 pages.
https://doi.org/10.1145/3106235

This work has been funded by NoemaLife SpA in the framework of the ConnectToLife project.
Authors' addresses: D. Marcheggiani, Institute for Logic, Language and Computation, Science Park 107, University of Amsterdam, 1098 XG Amsterdam, The Netherlands; email: d.marcheggiani@uva.nl; F. Sebastiani, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Via Giuseppe Moruzzi 1, 56124 Pisa, Italy; email: fabrizio.sebastiani@isti.cnr.it. The order in which the authors are listed is purely alphabetical; each author has given an equally important contribution to this work.

https://doi.org/10.1145/3106235

1 INTRODUCTION

Since the early 2010s 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), and Uzuner et al. (2011)) on information extraction from clinical documents, that is, 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. Many of these works are about methods based on supervised learning, that is, 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 for the clinical domain.[1] In applications of supervised learning, issues of quality in the training data may arise for different reasons:

(1) In several scenarios, it is often the case that the main goal of the coders (a.k.a. "annotators" or "assessors") that carry out the annotation work is fast turnaround and not annotation quality. An example is the (increasingly frequent) case in which annotation is performed via crowdsourcing on platforms such as, for example, Mechanical Turk, CrowdFlower, and so on[2] (Grady and Lease 2010; Snow et al. 2008).
(2) In many organizations, it is also the case that annotation work is carried out by junior staff (e.g., interns), since having it accomplished by senior employees would make costs soar. This is certainly the case in the clinical domain, where annotation is usually performed by medical students and/or trainees.
(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, which may negatively affect the quality of their annotation. For instance, in the clinical domain the authors of tagsets are usually senior clinical specialists, who usually do not then engage themselves in the coding phase.
(4) The data used for training the system may sometimes be old or outdated, with the annotations no longer reflecting the current meaning of the concepts. 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, the resulting annotations would be, to a certain extent, more reliable. Here, we define the *authoritative coder* (hereafter indicated as C_α) to be the coder who has annotated the test set (or the coder whose judgments we adhere to when evaluating the accuracy of the system[3]), while we define a non-authoritative coder (hereafter indicated as C_β) to simply be a coder different from the authoritative coder.

It is natural to expect the accuracy of an information extraction system to be lower if the training data have been annotated by C_β and higher if they have been annotated by C_α him- or herself.

[1] This statement refers to a specific notion of "quality of training data" to be discussed below; we are thus not making any claim concerning research that addresses possibly different notions of "quality of training data."
[2] https://www.mturk.com/, and http://crowdflower.com/.
[3] This clause also serves to characterize *accuracy* as the degree of coincidence between the annotations automatically generated by the system and the ones manually generated by the authoritative coder.

However, note that C_α is not necessarily more experienced, senior, or reliable than C_β. Rather, the fact that we expect higher accuracy if the training data have been annotated by C_α 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*), that is, 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, that is, 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, 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*. In the rest of this article, we will take the authoritative coder C_α to be *the coder whose annotations are to be taken as correct*, that is, considered as the "gold standard." C_α is thus 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).[4]

If the training data have been annotated by C_β, then, should it be independently re-annotated by C_α, we would be able to precisely measure this difference in reliability by measuring the *intercoder agreement* (via measures such as Cohen's kappa—see, for example, Artstein and Poesio (2008) and Di Eugenio and Glass (2004)) between the training data Tr as coded by C_α and the training data as coded by C_β. 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.

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).[5]

1.1 Our Contribution

In this article, 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 widely used supervised learners 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 coders, equally expert in the subject matter.[6] In other words, we try to answer the question: "What is the consequence of the fact that my training data are not sterling quality? (i.e., that the labels associated to

[4]In some organizations this authoritative coder may well be a fictional entity, for example, 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.

[5]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 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 that exploits synergies between training data expressed in different languages).

[6]Note that, as more fully explained in Section 3.2, we had no role in the annotation of the dataset; we thus take both the concept set and the dataset as given. Note also that our work entirely relies on preexisting data and that (for various reasons,

the training data are not the same as an authoritative annotator would have associated to them) What is the consequence of the fact that the coders who produced them are not authoritative? 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.

While our experiments are carried out on clinical data, our findings are of general interest, since no features unique to the clinical domain are used in processing the data.

The rest of the article 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.

2 RELATED WORK

2.1 Information Extraction from Clinical Documents

Many research works on information extraction from clinical documents rely on methods based on supervised learning, that is, 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, 2011), the 2013–2016 editions of the ShARe/CLEF eHealth IE-related tasks (Suominen et al. 2013; Kelly et al. 2014; Goeuriot et al. 2015; Névéol et al. 2016), the Semeval-2014 and Semeval-2015 Tasks "Analysis of Clinical Text" (Pradhan et al. 2014; Elhadad et al. 2015), and the Semeval-2016 Task "Clinical TempEval" (Bethard et al. 2016). In these shared tasks, the goal is to competitively evaluate (among others) information extraction tools that recognise mentions of various concepts of interest (e.g., mentions of diseases and disorders) as appearing in discharge summaries, electrocardiogram reports, echocardiograph reports, and radiology reports.

ranging from the lack of access to unannotated medical reports, to the lack of competence to annotate text according to medical concepts) we could not attempt to annotate new data for the purpose of this study.

2.2 Low-Quality Labels

As mentioned in the Introduction, in many fields where labelled data are used, obtaining high-quality (i.e., accurate) labels is expensive, since it requires the work of trained human assessors and of senior specialists who train and coordinate them. As a result, in many cases, one is willing to trade the quality of the labels obtained for a sizable reduction in the costs incurred for obtaining them. This has given rise to the notion of a *silver label*, that is, a label that is only probably accurate (as opposed to a gold label, which is—or we hypothesize to be—certainly accurate), and to the notion of a *silver standard*, that is, a labelled dataset where the labels are silver labels. Silver labels may be obtained either by speeding up the manual annotation work or by having a highly accurate automatic or semi-automatic process generate the labels (Kang et al. 2012; Rebholz-Schuhmann et al. 2010). There are two main uses for silver labels, that is, (a) as labels for training data (Kang et al. 2012) and (b) as labels for test data (Groza et al. 2013; Rebholz-Schuhmann et al. 2010). The latter use has been studied more than the former, since it is not confined to supervised learning environments; for instance, the TREC text retrieval evaluation campaign (Voorhees and Harman 2005) has been testing on silver standards since the early 1990s, since producing a gold standard of the size adequate for testing, say, Web search engines, is prohibitive.

2.3 Low-Quality Training Data and Prediction Accuracy

While the limits of using silver standards as test data have been studied fairly extensively, 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.[7] 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); its results are not easy to interpret, since also the test data (and not only the training data) used in the experiments are low quality. Yet another study, in which the goal was predicting the production levels of palm oil via a neural network, is Khamis et al. (2005); here, low training label quality is artificially generated by perturbing fixed percentages of training labels. This makes the results not very relevant to our study, which is instead concerned with naturally occurring label noise (in the form of labels attributed by a non-authoritative annotator).

Saarikoski et al. (2015) study the effects of imperfect training data quality on text classification accuracy. However, their notion of "data quality" is very different from ours. While in our work labels are categorical (i.e., a token either has a tag or not), in their work labels are soft (i.e., a given document may be labelled "irrelevant," "marginally relevant," "fairly relevant," or "highly relevant" to a given class), so, for example, an example "marginally relevant" to a given class counts as a low-quality training example while an example "highly relevant" to the class counts as a high-quality one. We avoid dealing with soft labels, since they are extremely rare in practice.

Kang et al. (2012) study the impact of using silver-labelled data, either alone or in conjunction to gold-labelled data, for training a "text chunker" (a recognizer of syntactically meaningful

[7]By "artificial noise" we mean simulated noise, that is, noise that is inserted by the experimented by perturbing the data for the sole purpose of testing *in vitro* the impact of noise on the process; by "naturally occurring noise," we mean noise which is not simulated, that is, is present in the original data.

multi-word units in natural language processing); differently from us, their silver labels are generated by an automatic process, while in our case they derive from the work of a human (non-authoritative) coder.

In the context of a biomedical information extraction task,[8] 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 not clear how well the type of noise they introduce models naturally occurring noise. See Berndt et al. (2015) for a further text classification study where artificial noise is inserted in the training data to explore how the accuracy of the resulting classifiers varies as a function of training data quality. 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 coding expertise of the two coders (it tackles the case of two coders with equal domain expertise, though).

2.4 Improving Training Data in Clinical IE

That training data quality is conducive to learning accurate models is intuitive. As a result, in several contexts a lot of effort is put into ensuring that annotation generates high-quality (i.e., correct) labels; this includes, for example, providing clear annotation guidelines to the annotators, conducting preliminary annotation exercises to align their understanding of the concepts whose mentions are sought in the documents, and so on. A good summary of best practices and rigorous methodologies for the construction of annotated corpora of clinical text can be found in Roberts et al. (2009).

An alternative route to ensuring label quality in the training items is *training data cleaning* (Esuli and Sebastiani 2009), whereby annotators are asked to check (and correct if needed) the labels of training data with the support of an algorithm which prioritises these training items according to how likely it is that the respective labels are wrong. In other words, while the techniques discussed in the previous paragraph try to ensure quality by affecting the annotation process, these techniques are applied after annotation has taken place already. Variants of this basic approach are *corrActive learning* (Nallapati et al. 2009) and *reverse active learning* (Nguyen and Patrick 2012), in both of which the operations of checking label correctness and retraining the system are interleaved in an iterative fashion.

When the quality of training data is not high and cannot be increased (e.g., due to the unavailability of humanpower for checking label correctness), one can attempt to make up for low

[8]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.

quality by increasing quantity. Since labelled data are scarce or expensive to obtain, a vast array of machine-learning techniques have been developed that try to leverage, for the training process, additional data that have not explicitly been annotated for the task at hand. This has spawned entire subfields of machine learning, such as *transfer learning* (Pan et al. 2012), *transductive learning* (Joachims 1999), and *semi-supervised learning* (Chapelle et al. 2006). Examples of these approaches in the clinical information extraction field are Wagholikar et al. (2013), which augments an existing, "local" training set by means of a "foreign" one to make up for the fact that the existing training data are scarce, and Roberts et al. (2015), which proposes pooling training data from different provenance by adding to the existing coarsely annotated data more finely annotated data.

3 METHODS

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 a *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.

Note that a t-unit can be labelled with zero, one, or several concepts at the same time; our task is thus an instance of *multi-label* IE. Following standard practice in multi-label supervised learning, we will deal with each $c_r \in C$ independently of the other concepts in C; we will thus drop the r subscript and, without loss of generality, deal with 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 $y_t \in \{c, \overline{c}\}$. While this "reduction to binary" does not allow us to exploit potential dependencies among different concepts in C, it considerably simplifies our treatment; the latter is the reason why the reduction to binary is the approach taken in the vast majority of works in the multi-label IE literature.

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, then this means that it has correctly identified the boundaries of the two mentions "Barack Obama" and "Hillary Clinton."[9]

[9]Note that the above notation is not able to represent "discontiguous mentions," that is, mentions containing gaps, and "overlapping mentions," that is, 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 article (the consequences on extraction accuracy of suboptimal training data quality) can be considered largely independent of the expressive power of the markup language.

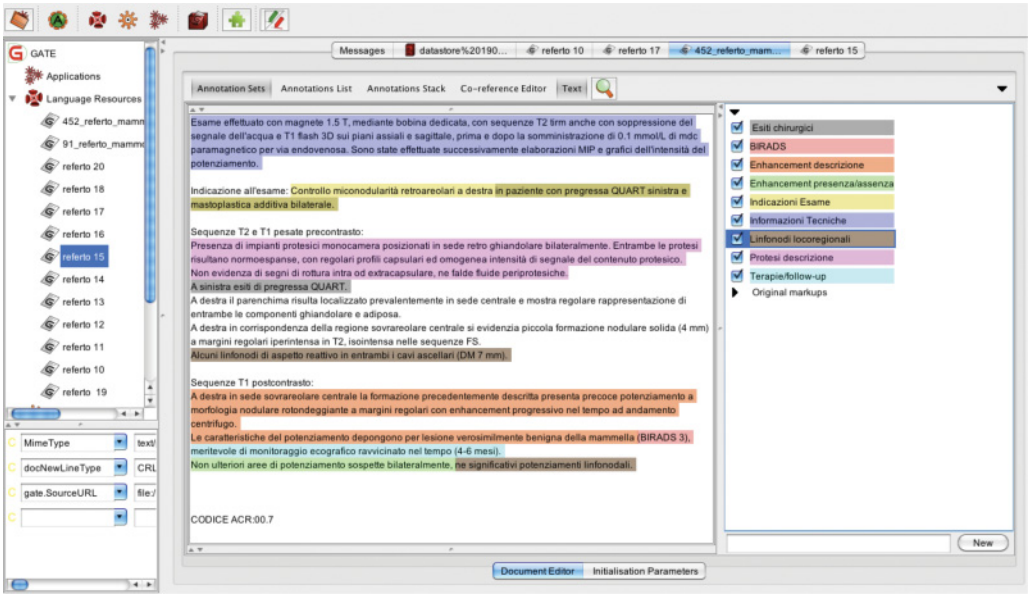

Fig. 1. A screenshot displaying a mammographic report automatically annotated according to the nine concepts of interest. The screenshot depicts the interface of the GATE system, which the two human coders have used for manually annotating the reports. Each of the nine colours corresponds to one of the concepts of interest, and each contiguous region of text highlighted with a colour is a mention of the corresponding concept.

3.2 Dataset

The dataset we have used to test the ideas discussed in the previous sections is the UmbertoI(RadRep) dataset discussed in Esuli et al. (2013), consisting of a set of 500 free-text mammography reports written (in Italian) by medical personnel of the Istituto di Radiologia of Policlinico Umberto I, Roma, IT. The dataset is annotated according to nine concepts relevant to the field of radiology and mammography: "Outcome of the BIRADS test" (BIR), "Technical Info" (ITE), "Indications obtained from the Exam" (IES), "Followup Therapies" (TFU), "Description of Enhancement" (DEE), "Presence/Absence of Enhancements" (PAE), "Outcomes of Surgery" (ECH), "Prosthesis Description" (DEP), and "Locoregional Lymph Nodes" (LLO). 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 (plus 16.33 separators).

Figure 1 displays a sample mammographic report automatically annotated according to the nine concepts of interest. This figure shows that this task is fairly different from many other concept extraction tasks in clinical IE, such as the extraction of drug names, drug dosages, names of pathologies, or their symptoms. Here, the spans to be annotated are longer (often taking up two or more sentences) and are characterized by a more irregular surface form (the mentions

Table 1. 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

may consist of sequences of full sentences but also of fragments of sentences or of a fragment of a sentence followed by a full sentence followed by another fragment of a sentence).

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) to align their understanding of the meaning of the concepts.

Table 1 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.[10]

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, 2012), in Charles Sutton's GRMM implementation,[11] and *hidden Markov support vector machines* (HM-SVMs) (Altun et al. 2003), in Thorsten Joachims's SVM^{hmm} implementation.[12] Both are supervised learning algorithms explicitly devised for *sequence labelling*, that is, 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).[13] LC-CRFs are members of the class 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,

[10]No other dataset is used in this article, since we were not able to locate a dataset of annotated clinical texts that (a) contains a sizeable amount of reports independently annotated by two coders c_1 and c_2 and (b) is publicly available. Note that also the dataset on which we have carried out our experiments has, unfortunately, not been made available by Policlinico Umberto I.

[11]http://mallet.cs.umass.edu/grmm/.

[12]http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html.

[13]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 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.

e.g., Tang et al. (2012) and 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.[14] In this work, we have used a set of features that 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 four "positional" features (Esuli et al. 2013, Section 3.3) that indicate in which half, third, 4fourth, or fifth, respectively, of the text the token occurs.

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 contributes to the calculation of the F_1 measure, in the sense that 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.

We compute F_1 across the entire test set, that is, we generate a single contingency table by putting together all t-units in the test set, irrespective of the document to which they belong. 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

[14]Note that only tokens, and not separators, are explicitly represented in vectorial form, the reasons being the same as those already discussed in Footnote 13.

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 him- or 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) and 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 $p = c$ (respectively, $t = c$) to mean that the predicted label (respectively, 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, that is, 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).[15] Those who conduct their evaluation at the mention level (e.g., Chapman and Dowling (2006)) find that they are unable to do so, since 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 other ICA measures that require the notion of a true negative to be defined.

3.5 Statistical Significance

To check whether differences in accuracy between different settings are statistically significant, we will use the *approximate randomization test* (ART) (Chinchor et al. 1993). In this test, the difference

[15]This would be possible also if we considered a text as a sequence of tokens (thus disregarding separators) and conducted the evaluation at the token level only. That is, the key aspect that allows the computation of Cohen's kappa is that atomic units of text, that do not overlap with each other, are used.

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).

3.6 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 article, we switch the roles of training set and test set, that is, use `Both(1)` or `Both(2)` as training set (since for the purpose of this article 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.[16]

Each of the two batches of experiments is composed of the following:

(1) An experiment using the authoritative setting, that is, 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, that is, 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, that is, test data annotated by C_α, and training data annotated in part by C_β (λ% 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 $\lambda \in \{10, 20, \ldots, 80, 90\}$ 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 $\lambda \in \{10, 20, \ldots, 80, 90\}$, each time with a different random such choice.

For each of the above train-and test experiments, 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

[16]The very fact that, in our experiments, we treat the two coders equally (e.g., Batch 1 and Batch 2 experiments use the very same protocol, and we give identical importance to their results) can be seen as using (i) the information that the two are equally expert radiologists and (ii) the lack of information on their relative expertise as coders (i.e., in the absence of information to the contrary, we assume them to have the same level of coding expertise). Had we had information that one was a more experienced radiologist than the other, or that one was a more experienced coder than the other, we might have treated the two batches differently (e.g., if we knew that Coder1 had substantially more coding expertise than Coder2, we would probably only consider the experiments in Batch 1, since it makes sense to have the more experienced coder be the one by whose judgment we must abide, that is, the one who annotates the test set).

Table 2. Extraction Accuracy for the Authoritative Setting ($\lambda = 0$) and Non-authoritative Setting ($\lambda = 100$), for the LC-CRFs and HM-SVMs Learners, and for Both Batches of Experiments (and for the Average across the Two Batches), Along with the Resulting Intercoder Agreement Values Expressed as $\kappa(\lambda)$

			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%)

Percentages indicate the loss in extraction accuracy resulting from moving from $\lambda = 0$ to $\lambda = 100$; F_1^μ and F_1^M are as defined at the end of Section 3.4.1.

denote it as $\kappa(\lambda)$; this value indicates the average intercoder agreement that derives by "corrupting" λ% of the documents in the training set, that is, by using for them the annotations performed by the non-authoritative coder.

For each of the above train-and test experiments, 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.

The experiments outlined above are discussed in Sections 4.1.1 to 4.2.1. In Section 4.2.2, we discuss a further experiment carried out via k-fold cross-validation exclusively on `Both(1)` and `Both(2)`, that is, using an experimental setting in which all data, although few, are doubly annotated; this eliminates any bias in the results that might potentially derive from `1-only` and `2-only` containing different documents.

4 RESULTS

Table 2 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 2 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 intercoder agreement $\kappa(\lambda)$ (as averaged across the two batches) is also indicated. Figure 3 plots instead precision and recall as a function of λ for the LC-CRFs experiments, while Figure 4 does the same for the HM-SVMs experiments.

4.1 The Authoritative Setting

We start the presentation of our results with a discussion of phenomena that can already be detected at the level of the authoritative setting, that is, with training set and test set completely annotated by the same person. This will set the stage for the discussion of what can instead be observed at the level of the non-authoritative and of the partially authoritative settings, which are the main focus of this article.

4.1.1 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

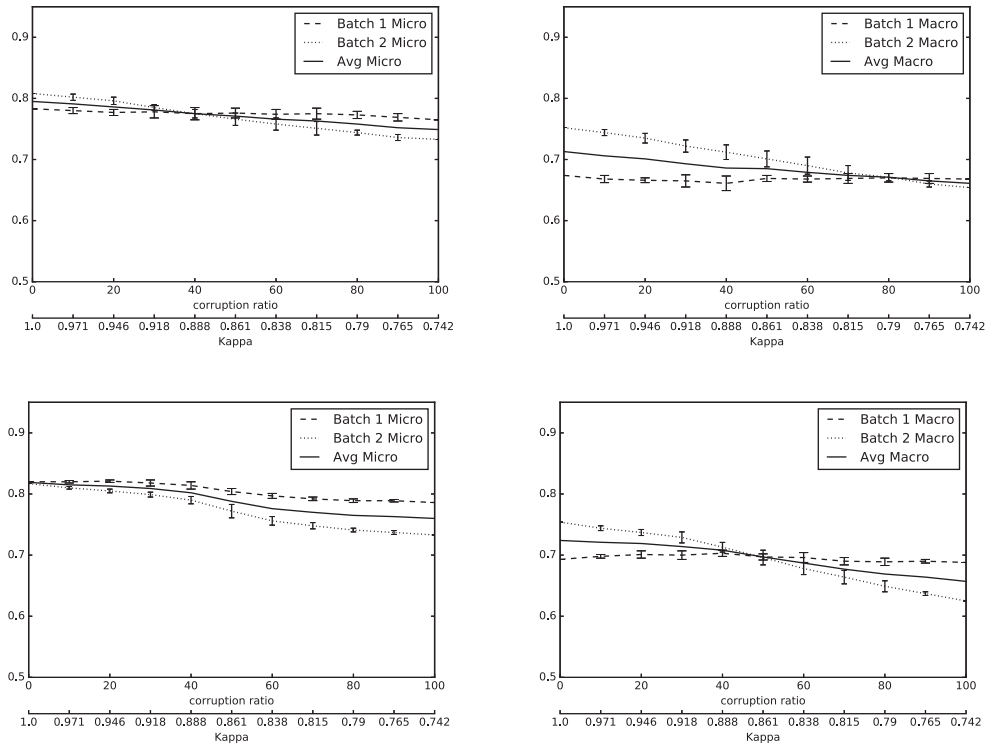

Fig. 2. Microaveraged F_1 (left) and macroaveraged F_1 (right) on the UmbertoI(RadRep) dataset as a function of the fraction λ of the training set that is annotated by C_β instead of C_α ("corruption ratio"), using LC-CRFs (top) and HM-SVMs (bottom) as learning algorithms. The dashed line represents 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 $\lambda \in \{10, 20, \ldots, 80, 90\}$, the standard deviation across the 10 runs deriving from the 10 random choices of $corr_\lambda(Tr)$.

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, which means that microaveraged effectiveness scores tend to be higher, too. 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; this means that macroaveraged effectiveness scores tend to be lower. See Debole and Sebastiani (2005, pp. 591–593) for a thorough discussion of this point in a text classification context.

4.1.2 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 for both evaluation measures (F_1^μ and F_1^M); for example, 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 3 and 4. 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

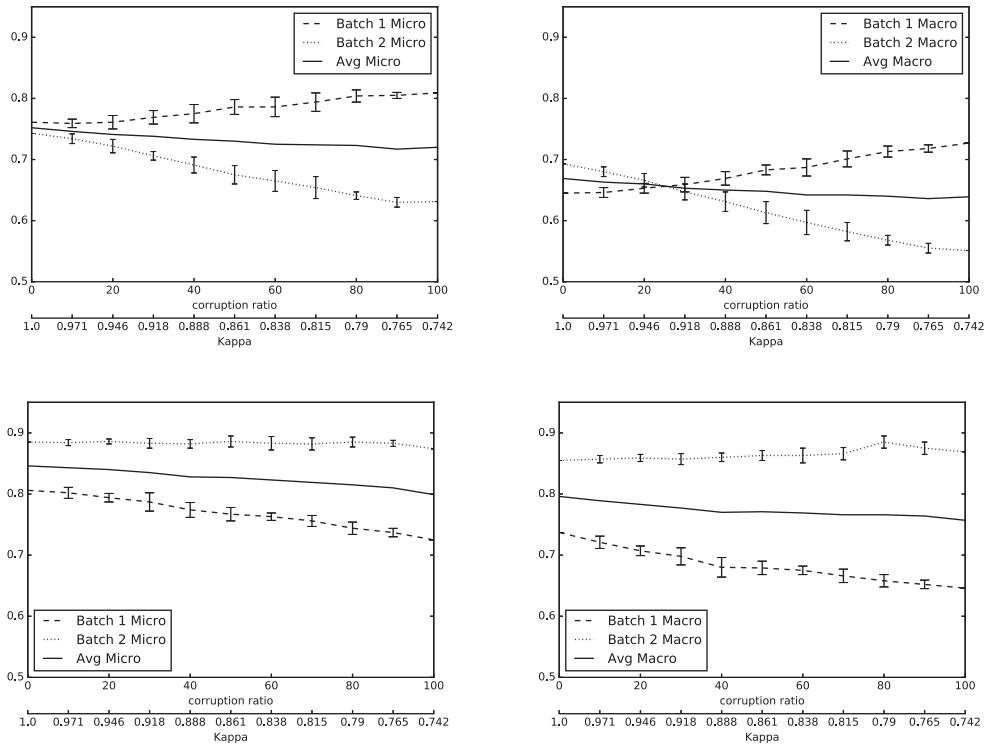

Fig. 3. Microaveraged (left) and macroaveraged (right) precision (top) and recall (bottom) on the Umbertol(RadRep) dataset as a function of the fraction λ of the training set that is annotated by C_β instead of C_α ("corruption ratio"), using LC-CRFs as a learning algorithm.

that follows, most insights obtained from the LC-CRFs experiments are qualitatively confirmed by the HM-SVMs experiments and vice versa.

4.1.3 Coder1 Generates Less Accuracy Than Coder2. A third fact that may be noted (from Table 2) 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 have $F_1^M = 0.693$ for Coder1 and $F_1^M = 0.754$ for Coder2), while the difference is less clear-cut 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.

Why do the two coders bring about this difference in accuracy? Possible explanations might be that the documents in `2-only` are "easier" to code automatically than those in `1-only` or that the distributions of `Both(1)` and `1-only` are less similar to each other than the distributions of `Both(2)` and `2-only`. However, both hypotheses will be ruled out by the experiments discussed in Section 4.2.2. There are instead two other possible explanations for this fact that our experiments will not rule out; we describe them in the next paragraphs.

The first possible explanation is that Coder2 might simply be more self-consistent in his or her annotation style than Coder1. To check whether this hypothesis is plausible, we have performed

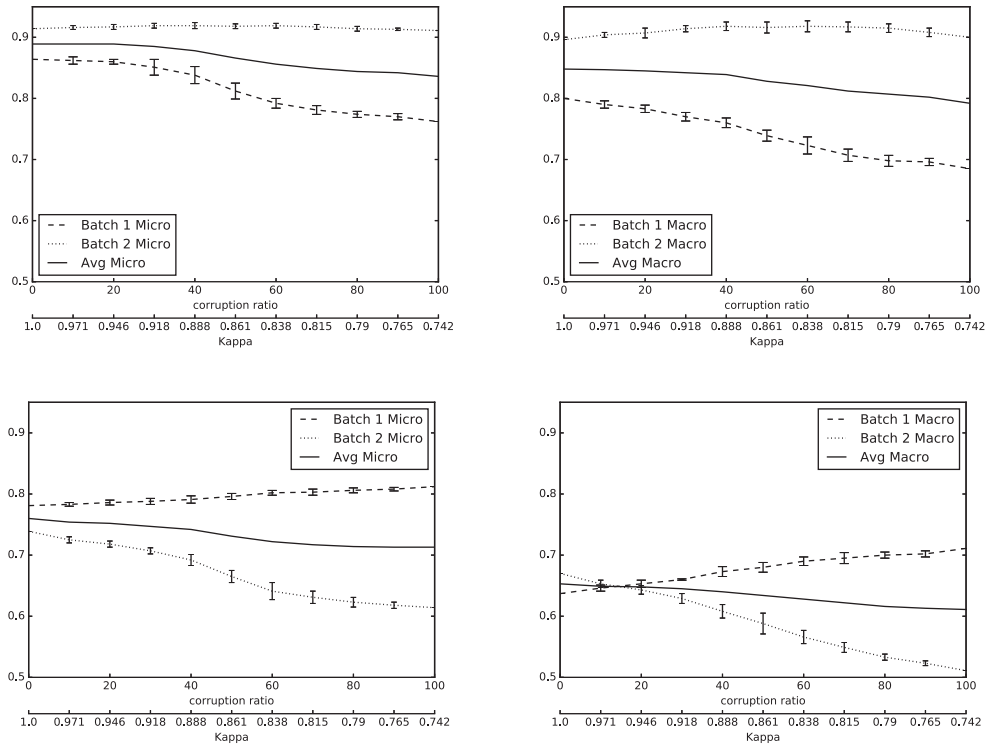

Fig. 4. Microaveraged (left) and macroaveraged (right) precision (top) and recall (bottom) on the Umbertol(RadRep) dataset as a function of the fraction λ of the training set that is annotated by C_β instead of C_α ("corruption ratio"), using HM-SVMs as a learning algorithm.

four k-fold cross-validation (k-FCV) experiments (on `Both(1)` and `Both(2)` and for LC-CRFs and HM-SVMs, in all combinations) using $k = 20$. Focusing on the documents in `Both`, thus leaving aside all documents that are not doubly annotated, allows us to rule out possible explanations having to do with the difference among the test documents contained in `1-only` and `2-only`. Intuitively, a higher accuracy value resulting from a k-FCV test might indicate 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 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 3. From this table, we can see that the accuracy on `Both(2)` is substantially higher than the one obtained on `Both(1)`. This might indicate that Coder2 is indeed more self-consistent than Coder1, which might be an explanation of the higher levels of accuracy obtained on the dataset annotated, for both training and test, by Coder2.

The second possible explanation is that, since Coder2 annotates more tokens and more mentions as belonging to the concept of interest, this has the effect of generating more training data,

Table 3. 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

and this usually entails higher accuracy. In fact, as is evident from Table 1, Coder2 annotates, as instances of the concepts of interest, more mentions (+15.7%) and also more tokens per mention (+15.6%) than Coder1; relative to each other, Coder1 is thus an *underannotator* while Coder2 is an *overannotator*.[17] Indeed, the results of the k-FCV experiments reported in the previous paragraph are entirely consistent with this second explanation, too.

Deciding which of the two explanations is the most plausible is not easy. To do this, we should selectively remove, from Coder2's annotations, a number of mentions and tokens such that the remaining ones are equal in number to Coder1's; at this point, if Coder2 still generates high accuracy than Coder1, then superior self-consistency (and not higher amounts of training data) is the explanation for the observed phenomenon. But it is evident that this selective removal cannot be performed without introducing bias against Coder2.[18] Therefore, we will not attempt to precisely determine the exact reason why Coder2 generates higher accuracy than Coder1; luckily enough, this will not negatively impact the analysis we will carry out in the next sections.

4.2 The Partially Authoritative and the Non-Authoritative Settings

We now discuss the results of our experiments using the partially authoritative and the non-authoritative settings. These settings are the ones in which the quality of training data is sub-optimal and are thus the main focus of this article.

4.2.1 Overannotation and Underannotation. The most interesting fact we may observe in the partially authoritative and the non-authoritative settings is that accuracy as a function of the corruption ratio varies much less for Batch1 than for Batch2, since for the latter we witness a much more substantial drop in going from $\lambda = 0$ to $\lambda = 100$. We conjecture that this may be due to the fact, noted in the previous paragraph, that Coder1 is an underannotator and Coder2 is an overannotator; the rest of this subsection will be devoted to explaining the rationale of this conjecture.

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 3 and 4 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

[17]Note that in this article, we use the term "underannotator" not in an absolute sense but in a relative sense, that is, we do not mean that this person annotates too little but that he or she annotates less than the other person. Same for "overannotator."

[18]To bring Coder2's annotations down to the number and size of Coder1's, it is not clear which mentions we should remove, and it is not clear which tokens we should remove from the remaining mentions. If we removed random mentions, and random tokens from the remaining mentions, then it is almost certain that the remaining set of mentions would not resemble anything coherent. For instance, assume that the same sentence occurs in two different documents and that Coder2 has (coherently) annotated both sentences under concept c_r; if we remove one of the two annotations but not the other, then we introduce inconsistency in what would otherwise be a consistent set of annotations. Same if the two sentences annotated in the two documents are not identical but just similar in meaning.

Table 4. 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 $\lambda = 100$

	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

Results are reported for both learners (LC-CRFs and HM-SVMs), both batches, and both evaluation measures (F_1^μ and F_1^M).

(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 2 (and in Table 2 too), the drop in F_1 resulting from moving to $\lambda = 0$ to $\lambda = 100$ is very limited. Figures 3 and 4 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 to the combined effect of a decrease in true positives and a decrease in false positives); as a result, as shown in Figure 2 (see also Table 2), the drop in F_1 resulting from moving to $\lambda = 0$ to $\lambda = 100$ is much more substantial than for Batch1.

The results of our statistical significance tests, carried out via the approximate randomization test described in Section 3.5, are reported in Table 4. These results essentially confirm the observations above, that is, that *in Batch1 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 Batch2.

4.2.2 Fivefold Cross-Validation Experiments. The experiments we have discussed in Section 4.2.1 might be considered problematic, because the differences in the performance obtained on `1-only` and `2-only` could in principle be attributed to the fact that `1-only` and `2-only` contain different documents.

To address this potential concern, we have run another set of experiments in which we do away with the documents in `1-only` and `2-only` and focus on the documents in `Both`. More specifically, we have run a fivefold cross-validation (5FCV) experiment by (a) splitting `Both(1)` in five folds $\texttt{Both(1)}_1, \ldots, \texttt{Both(1)}_5$ of equal size, and (b) running, for each of the $\texttt{Both(1)}_i$, one experiment in which $\texttt{Both(1)}_i$ is the test set and either $\bigcup_{j \neq i} \texttt{Both(1)}_j$ or $\bigcup_{j \neq i} \texttt{Both(2)}_j$ is the training set. Below we refer to this experiment as Batch1; we also run a Batch2 experiment, in which we split in five folds `Both(2)` instead of `Both(1)` and then proceed analogously to Batch1.

This experimental setting is conceptually identical to the one we have discussed in the previous sections, the only difference being the fact that the dataset used here entirely consists of doubly annotated documents. This latter experimental setting has advantages and disadvantages with respect to the one we had used previously. The advantage is that we know that any difference in accuracy between the two trained systems is a result of the annotations on which the systems were trained and not of the test documents; the disadvantage is that the dataset on which the experiment is performed is, overall, smaller (191 documents instead of 500).

The results are displayed in Figures 5, 6, and 7, which are the 5FCV analogues of Figures 2, 3, and 4, respectively. As revealed by a visual inspection of these figures, these 5FCV experiments

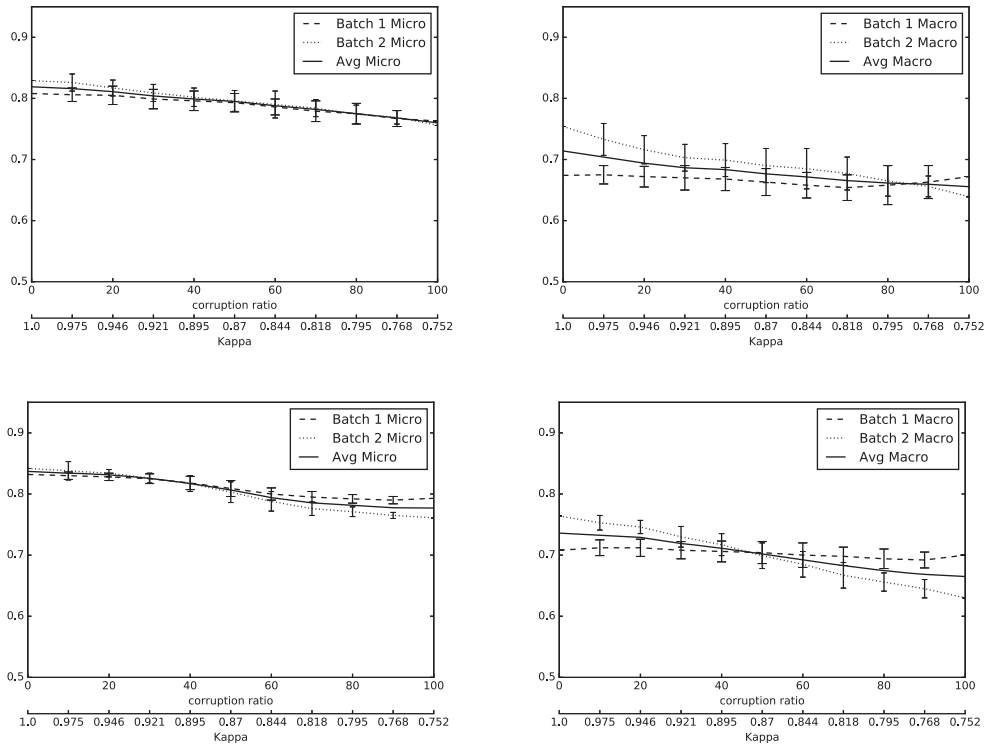

Fig. 5. Microaveraged F_1 (left) and macroaveraged F_1 (right) on the UmbertoI(RadRep)(191) dataset as a function of the fraction λ of the training set that is annotated by C_β instead of C_α ("corruption ratio"), using LC-CRFs (top) and HM-SVMs (bottom) as learning algorithms. The dashed line represents 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 $\lambda \in \{10, 20, \ldots, 80, 90\}$, the standard deviation across the 10 runs deriving from the 10 random choices of $corr_\lambda(Tr)$.

confirm the results of the previous experiments, in that each pair of plots (consisting of one of the 12 plots in Figures 2, 3, and 4 and its analogue in Figures 5, 6, and 7) qualitatively exhibit the same behaviour. For instance, in the bottom right plot of Figure 2 and in the bottom right plot of Figure 5, both representing the trends of F_1^M in the HM-SVMs experiments, effectiveness is higher for Batch1 than for Batch2 for $\lambda = 0$, is the other way around for $\lambda = 100$, and the two batches reach the same effectiveness around $\lambda = 50$. All the 12 pairs of plots (with the possible exception of the top left plots of Figures 2 and 5, representing the trends of F_1^μ in the LC-CRFs experiments and whose similarity is less marked) exhibit such qualitative similarity, which essentially confirms the conclusions we had drawn in the preceding sections.

4.2.3 Caveats. The experiments discussed in this article 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 10, we have not been able to locate any dataset that has the required characteristics (that is, contains a sizeable amount of doubly annotated documents) and is also publicly available.

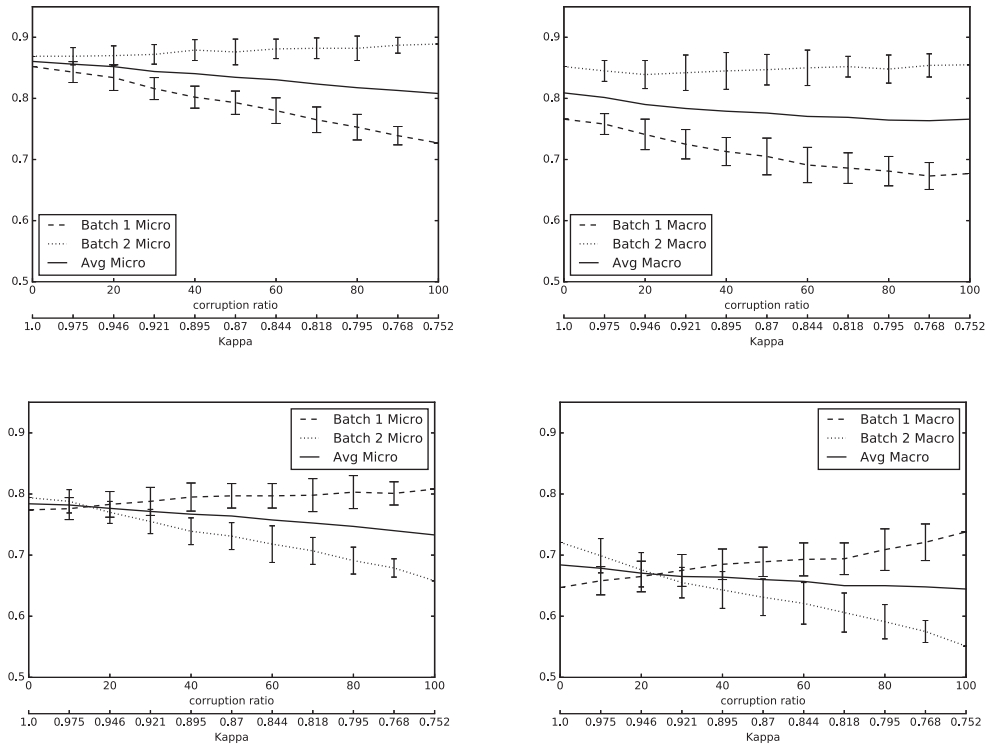

Fig. 6. Microaveraged (left) and macroaveraged (right) precision (top) and recall (bottom) on the UmbertoI(RadRep)(191) dataset as a function of the fraction λ of the training set that is annotated by C_β instead of C_α ("corruption ratio"), using LC-CRFs as a learning algorithm.

(2) The dataset used here is representative of only a specific type of imperfect training data quality, that is, the one deriving from the fact that the training data were annotated by a coder different (albeit with equal domain expertise) 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 4.2.1, 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 authoritative 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

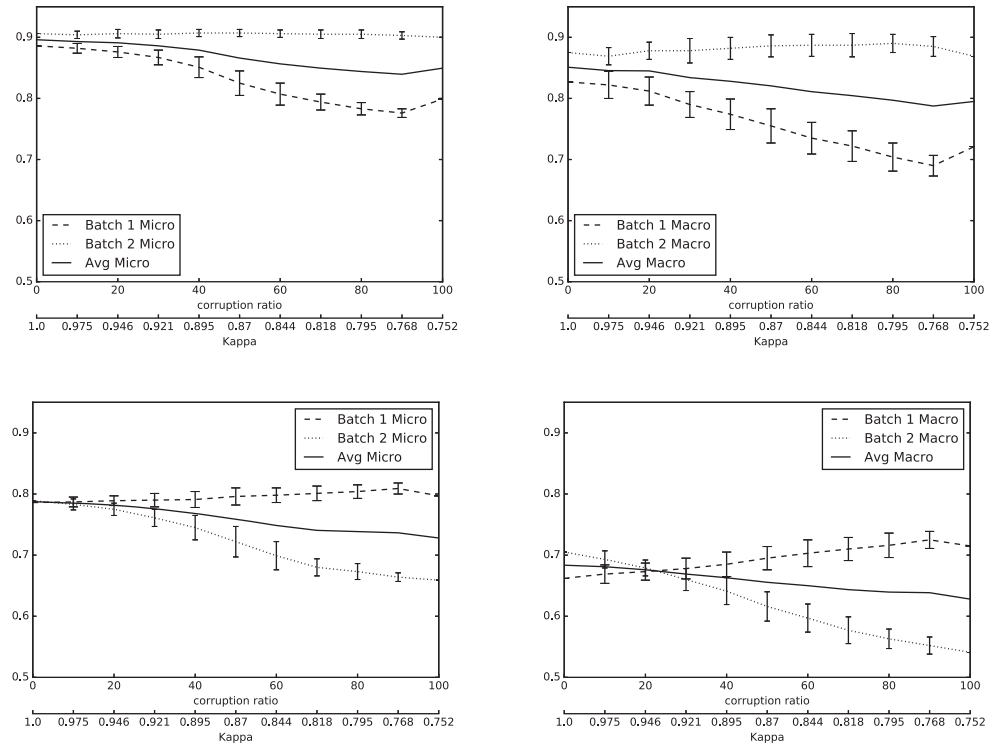

Fig. 7. Microaveraged (left) and macroaveraged (right) precision (top) and recall (bottom) on the UmbertoI(RadRep)(191) dataset as a function of the fraction λ of the training set that is annotated by C_β instead of C_α ("corruption ratio"), using HM-SVMs as a learning algorithm.

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 his or her level of (coding or domain) 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 at the token (word) level according to nine concepts of interest by two coders equally expert in the subject matter, 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 additional 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 having non-authoritative coders (typically, low-cost annotation workers) label the training data.

This article 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. One limit of this study is that it

only concerns coders who are equally expert in the subject matter (radiology, in our case); it would be interesting to carry out analogous studies tackling the situation in which the two coders have different levels of domain expertise (typically, with the coder who annotates the training data being less expert than the one who annotates the test data), since this situation may well be representative of a realistic scenario. Unfortunately, carrying out such a study requires a correspondingly annotated dataset, which would need to be annotated on purpose by medical personnel.

As repeatedly mentioned in this article, a further, related 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 publicly available medical datasets that contain (at least a subset of) textual records independently labelled by two different coders (Coder1 and Coder2); 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 (see also Footnote 10). 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 the existing datasets available to the scientific community.

ACKNOWLEDGMENTS

Thanks to the Istituto di Radiologia, Politecnico Umberto I, Roma, IT, for making available to us the dataset used in this work; to Giulia Chiaruzzi and Cristiano Querzé for assistance in obtaining it; to Gianpiero Camilli, Michele Carenini, and Davide Distefano for encouraging this work; and to Emanuele Pianta, Christian Girardi, and Roberto Zanoli for giving us access to their TextPro system.

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'03)*. 3–10.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Comput. Linguist.* 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. *Comput. Linguist.* 37, 4 (2011), 699–725.

Donald J. Berndt, James A. McCart, Dezon K. Finch, and Stephen L. Luther. 2015. A case study of data quality in text mining clinical progress notes. *ACM Trans. Manage. Inf. Syst.* 6, 1 (2015).

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. 1052–1062.

Olivier Chapelle, Bernard Schölkopf, and Alexander Zien (Eds.). 2006. *Semi-Supervised Learning*. The MIT Press, Cambridge, MA.

Wendy W. Chapman and John N. Dowling. 2006. Inductive creation of an annotation schema for manually indexing clinical conditions from emergency department reports. *J. Biomed. Inform.* 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). *Comput. Linguist.* 19, 3 (1993), 409–449.

Franca Debole and Fabrizio Sebastiani. 2005. An analysis of the relative hardness of Reuters-21578 subsets. *J. Am. Soc. Inf. Sci. Technol.* 56, 6 (2005), 584–596.

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

Noémie Elhadad, Sameer Pradhan, Sharon Gorman, Suresh Manandhar, Wendy Chapman, and Guergana Savova. 2015. SemEval-2015 task 14: Analysis of clinical text. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval'15)*. Denver, US, 303–310.

Andrea Esuli, Diego Marcheggiani, and Fabrizio Sebastiani. 2013. An enhanced CRFs-based system for information extraction from radiology reports. *J. Biomed. Inform.* 46, 3 (2013), 425–435.

Andrea Esuli and Fabrizio Sebastiani. 2009. Training data cleaning for text classificatssion. In *Proceedings of the 2nd International Conference on the Theory of Information Retrieval (ICTIR'09)*. Cambridge, UK, 29–41.

Andrea Esuli and Fabrizio Sebastiani. 2010. Evaluating information extraction. In *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation (CLEF'10)*. Padova, IT, 100–111.

Andrea Esuli and Fabrizio Sebastiani. 2013. Improving text classification accuracy by training label cleaning. *ACM Trans. Inform. Syst.* 31, 4 (2013), Article 19.

Lorraine Goeuriot, Liadh Kelly, Hanna Suominen, Leif Hanlen, Aurélie Névéol, Cyril Grouin, João R. M. Palotti, and Guido Zuccon. 2015. Overview of the CLEF eHealth evaluation lab 2015. In *Proceedings of the 6th International Conference of the CLEF Association (CLEF'15)*. 429–443.

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*. 172–179.

Tudor Groza, Anika Oellrich, and Nigel Collier. 2013. Using silver and semi-gold standard corpora to compare open named entity recognisers. In *Proceedings of the 7th IEEE International Conference on Bioinformatics and Biomedicine (BIBM'13)*. 481–485.

Sonal Gupta, Diana L. MacLean, Jeffrey Heer, and Christopher D. Manning. 2014. Induced lexico-syntactic patterns improve information extraction from online medical forums. *J. Am. Med. Inf. Assoc.* 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'08)*.

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'09)*, Vol. II.

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. *J. Am. Med. Inf. Assoc.* 18, 5 (2011), 601–606.

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning (ICML'99)*. 200–209.

Siddhartha Jonnalagadda, Trevor Cohen, Stephen Wu, and Graciela Gonzalez. 2012. Enhancing clinical concept extraction with distributional semantics. *J. Biomed. Inform.* 45, 1 (2012), 129–140.

Ning Kang, Erik M. van Mulligen, and Jan A. Kors. 2012. Training text chunkers on a silver standard corpus: Can silver replace gold? *BMC Bioinform.* 13, 17 (2012).

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'14)*. 172–191.

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

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to its Methodology*. Sage, Thousand Oaks, CA.

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'01)*. 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'08)*. 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 (IHI'12)*. 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. Geissbuhlerand C. Kulikowski (Eds.). Schattauer Publishers, Stuttgart, DE, 128–144.

Ramesh Nallapati, Mihai Surdeanu, and Christopher Manning. 2009. CorrActive learning: Learning from noisy data through human interaction. In *Proceedings of the IJCAI 2009 Workshop on Intelligence and Interaction*.

Aurélie Névéol, Kevin Bretonnel Cohen, Cyril Grouin, Thierry Hamon, Thomas Lavergne, Liadh Kelly, Lorraine Goeuriot, Grégoire Rey, Aude Robert, Xavier Tannier, and Pierre Zweigenbaum. 2016. Clinical information extraction at the CLEF eHealth evaluation lab 2016. In *Proceedings of the Working Notes of CLEF 2016—Conference and Labs of the Evaluation Forum*. Évora, PT, 28–42.

Dung Nguyen and Jon D. Patrick. 2012. Reverse active learning for optimising information extraction training production. In *Proceedings of the 25th Australasian Joint Conference on Artificial Intelligence (AI'12)*. Sydney, AU, 445–456.

Weike Pan, Erheng Zhong, and Qiang Yang. 2012. Transfer learning for text mining. In *Mining Text Data*, Charu C. Aggarwaland ChengXiang Zhai (Eds.). Springer, Heidelberg, 223–258.

Jon Patrick and Ming Li. 2010. High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge. *J. Am. Med. Inf. Assoc.* 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'14)*. 54–62.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence (Eds.). 2009. *Dataset Shift in Machine Learning*. The MIT Press, Cambridge, MA.

Dietrich Rebholz-Schuhmann, Antonio Jimeno-Yepes, Erik M. van Mulligen, Ning Kang, Jan A. Kors, David Milward, Peter T. Corbett, Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. 2010. The CALBC silver standard corpus. *J. Bioinform. Comput. Biol.* 8, 1 (2010), 163–179.

Angus Roberts, Robert Gaizauskas, Mark Hepple, George Demetriou, Yikun Guo, Ian Roberts, and Andrea Setzer. 2009. Building a semantically annotated corpus of clinical texts. *J. Biomed. Inform.* 42, 5 (2009), 950–966.

Kirk Roberts, Sonya E. Shooshan, Laritza Rodriguez, Swapna Abhyankar, Halil Kilicoglu, and Dina Demner-Fushman. 2015. The role of fine-grained annotations in supervised recognition of risk factors for heart disease from EHRs. *J. Biomed. Inform.* 58, Suppl:S111-9 (2015).

Donald F. Rossin and Barbara D. Klein. 1999. Data errors in neural network and linear regression models: An experimental comparison. *Data Quality J.* 5, 1 (1999).

Jyri Saarikoski, Henry Joutsijoki, Kalervo Järvelin, Jorma Laurikkala, and Martti Juhola. 2015. On the influence of training data quality on text document classification using machine learning methods. *Int. J. Knowl. Eng. Data Min.* 3, 2 (2015), 143–169.

Claude Sammut and Michael Harries. 2011. Concept drift. In *Encyclopedia of Machine Learning*, Claude Sammutand Geoffrey I. Webb (Eds.). Springer, Heidelberg, 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'06)*. 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'08)*. 254–263.

Weiyi Sun, Anna Rumshisky, and Özlem Uzuner. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *J. Am. Med. Inf. Assoc.* 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'13)*. 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, MA, 93–127.

Charles Sutton and Andrew McCallum. 2012. An introduction to conditional random fields. *Found. Trends Mach. Learn.* 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 Association for Computational Linguistics (ACL/COLING'06)*. 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'12)*. 13–20.

Manabu Torii, Kavishwar Wagholikar, and Hongfang Liu. 2011. Using machine learning for concept extraction on clinical documents from multiple data sources. *J. Am. Med. Inf. Assoc.* 18, 5 (2011), 580–587.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.* 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. *J. Am. Med. Inf. Assoc.* 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. *J. Am. Med. Inf. Assoc.* 18, 5 (2011), 552–556.

Ellen M. Voorhees and Donna K. Harman (Eds.). 2005. *TREC. Experiment and Evaluation in Information Retrieval*. The MIT Press, Cambridge, MA.

Kavishwar B. Wagholikar, Manabu Torii, Siddhartha Jonnalagadda, and Hongfang Liu. 2013. Pooling annotated corpora for clinical concept extraction. *J. Biomed. Sem.* 4 (2013), Article 3.

Martin J. Wainwright and Michael I. Jordan. 2008. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.* 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*. 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'13)*. 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'00)*. 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)*. 802–806.

Received March 2015; revised March 2017; accepted June 2017