©1993 The Institute of Mind and Behavior, Inc.
The Journal of Mind and Behavior
Summer 1993, Volume 14, Number 3
Pages 223–234
ISSN 0271–0137

Self-talk and Self-awareness: On the Nature of the Relation

Alain Morin
Memorial University of Newfoundland

This article raises the question of how we acquire self-information through self-talk—i.e., of how self-talk mediates self-awareness. It is first suggested that two social mechanisms leading to self-awareness could be reproduced by self-talk: engaging in dialogues with ourselves, in which we talk to fictive persons, would permit an internalization of others' perspectives; and addressing comments to ourselves about ourselves, as others do toward us, would allow an acquisition of self-information. Secondly, it is proposed that self-observation (self-awareness) is possible only if there exists a distance between the individual and any potentially observable self-aspect; self-talk, because it conveys self-information under a different form (i.e., words), would create a redundancy—and with it, a wedge—within the self.

Each and every day of our waking life, at every hour, and even, arguably, at least once a minute, we talk to ourselves. No doubt, this makes self-talk (or internal dialogue, inner speech) a rather important psychological activity. But not only do we use inner speech frequently: it also shapes our thoughts, feelings and behaviors in a great variety of ways. For instance, it has been shown that inner speech plays a decisive role in self-regulation (Luria, 1959, 1961; Meichenbaum, 1976, 1977; Vygotsky, 1934/1962; Zivin, 1979), in problem solving (Harris, 1986; Hunsley, 1987; Roberts, 1979; Sokolov, 1972), and in planning (Meacham, 1979; Morin, 1988). A large number of pathologies, ranging from anxiety (Cacioppo, Glass, and Merluzzi, 1979; Meichenbaum and Butler, 1980) to depression (Beck, Brown, Steer, Eidelson, and Riskind, 1987), and including gambling (Gaboury and Ladouceur, 1989; Ladouceur, Gaboury, and Duval, 1988), guilt (Firestone, 1987), agoraphobia (Chambless, Caputo, Bright, and Gallagher, 1984), and many others, are mediated by mal-

I wish to thank James Everett, François Tournier, J. Michael Russell, and Gladys Symons for their helpful comments on earlier versions of this article. Requests for reprints should be sent to Alain Morin, 823 Nouvelle-Orléans, Ste-Foy (Québec), Canada G1X 3J4.

adaptive self-talk. An asymmetry between positive and negative self-statements seems to exist, where negative thoughts would have a greater functional impact (Schwartz, 1986); this effect could be even more pronounced with people having psychological problems (Clark and Channon, 1987). Studies have also linked self-talk to alcohol consumption (Oei and Young, 1987), performance in sport (Highlen and Bennett, 1983), reading (Yaden, 1984), effectiveness of counseling (Fuqua, Newman, Anderson, and Johnson, 1986; Kline, 1983; Kurpius, Benjamin, and Morran, 1985; Uhlemann, Lee, and Hiebert, 1988), and creativity (Bibler, 1987). It is thus my opinion that what we say to ourselves represents a cornerstone in the way we think and act.

We can also be self-aware, that is, we can take ourselves as the object of our own attention (Duval and Wicklund, 1972). This ability, unique to humans and some higher primates (see Gallup, 1985), also represents a cornerstone in the way we think and act because self-reflection shapes our feelings and behaviors in a variety of ways. To illustrate this point, just consider the following examples. Research has shown that highly self-aware individuals, in comparison to low self-aware individuals, perceive the content of their subjective experience more intensely and more acutely (Scheier and Carver, 1977, 1981), react more strongly to social rejection (Fenigstein, 1979), and know themselves better (Carver and Scheier, 1981; Turner, 1976). Self-awareness motivates self-evaluation and self-regulation: self-aware individuals find themselves in a better position than non self-aware individuals to compare their current states with their standards or goals and to try to conform to them (Duval and Wicklund, 1972; Scheier and Carver, 1988; see also Pinard, 1992). Also, self-awareness affects, among many social behaviors, conformity and attitude change (Froming and Carver, 1981), and compulsive self-awareness mediates psychopathology (especially depression and social anxiety) [Buss, 1980; Ingram, 1990]. Another important consequence of self-awareness is the capacity it brings to make inferences about others' mental states (Gallup and Suarez, 1986)—a capacity which, in turn, shapes our social relationships. Obviously, self-awareness represents a central psychological parameter.

Self-talk and self-awareness clearly are, in the author's opinion, at the heart of many psychological phenomena. The existence of a link between inner speech and self-awareness has been recently proposed (Morin and Everett, 1990a). When we ask: How do we learn about ourselves? (or: How do we develop a self-concept?), three sources of self-information can readily be identified: the social world, one's physical environment (i.e., self-reflecting devices such as mirrors, photographs and videotape recordings), and the self. We have a rather sophisticated knowledge of how the social world, for instance, brings about self-information. According to Mead (1912/1964, 1934, 1982; see also Meltzer, 1991; Natsoulas, 1985), being confronted with different ways of thinking, feeling and behaving would first allow the individual to perceive that he or she is endowed with unique qualities, and then would motivate him or her to take others' perspectives to gain an objective vision of himself or herself. Cooley (1902; see also McCall, 1977) proposed that we learn about ourselves by being repeatedly exposed to verbal comments—or appraisals—others emit about us.

But what about the self? Self-awareness, so it seems, is a rather mysterious phenomenon. When the individual is self-aware, he or she becomes his or her own source of self-information. But an intriguing question here is: What *really* takes place when the individual *examines*, *analyzes* himself or herself, *reflects* or *focuses* on himself or herself? As Gibbons points out (1990, p. 250), little is known about what happens cognitively when the individual is self-aware.

In answer to the above-mentioned question, it can be proposed that when self-aware, the individual, more often than not, *talks to himself or herself*. In other words, self-talk would be a mediator of self-awareness—an important tool involved in the acquisition of self-information.

The hypothesis of the existence of a link between self-talk and self-awareness has been shown to be both logically and empirically plausible (see Morin, 1992; Morin, Everett, Turcotte, and Tardif, 1993). What is less clear and of interest here concerns the *nature* of this relation. Of course, we can easily conceive of self-awareness *activating* (i.e., causing) a self-conversation. However, perceiving the relation in the other direction, where self-talk would *mediate* self-awareness, raises perhaps more intriguing—and more interesting—questions: How does self-talk mediate self-awareness? How do we learn about ourselves through self-talk? These questions are the focus of the present article.

Reproduction by Self-talk of Psychosocial Mechanisms Responsible for the Acquisition of Self-information

I mentioned above that one possible source of self-information is the social world. I would suggest that social mechanisms proposed by social interactionists (or *inter*-personal modes of acquisition of self-information) could be reproduced by cognitive processes (or *intra*-personal modes of acquisition of self-information), and especially by self-talk, where conversations with ourselves would permit an internalization of others' perspectives [Mead] and a replication of comments emitted by others [Cooley] (see Morin and Everett, 1990a, 1990b; Morin and DeBlois, 1989; see also MacKay, 1979). Luria (1978) proposed that the organization of the brain's "higher" and more "noble" functions has been shaped by the social environment in which it evolved. I believe that the social world is a necessary, but not a sufficient condition for the emergence of self-awareness. We know for instance that a motivation to communicate with others, although rooted in the social environment, needs to be mediated by linguistic processes in order to manifest itself effectively. The same could be true for self-awareness: once initiated by the social environment, it could be argued that this initial social movement should then be taken over and extended by cognitive processes (Morin and Everett, 1990b). Moreover, if we were only to have social feedback as a source of self-information, we could hardly acquire self-information *outside social situations* (Morin and DeBlois, 1989).

Now, since the social milieu brings self-information, and self-talk could reproduce the mechanisms by which this social milieu conveys self-information, it is reasonable to assume that self-talk mediates self-awareness. Let me examine and illustrate this idea.

Creating an Objective Point of View by Self-talk

The first inter-personal mode of acquisition of self-information I identified was Mead's idea that confrontations with others force the individual to take others' perspectives in order to gain an objective point of view regarding himself or herself. Once in this position, the individual can acquire self-information.

Before going any further, it should be pointed out that this mechanism, although psychosocial in nature, must surely be mediated by some cognitive process—by inner speech, for example—to open into an effective acquisition of self-information. Just consider the following steps involved in the social mechanism under scrutiny. The individual must first observe others. He or she might say to himself or herself "So, that's what these people think!" Then, self-observation must take place so as to realize that there exists an incongruence between what the individual perceives of others and what he or she perceives of himself or herself: "But I've never thought that way!" And finally, the individual must identify the difference in order to perceive a particular self-aspect. He or she might engage in the following internal dialogue: "These people conceive of (this thing, this problem) in terms of *x*, whereas I see (the same thing or problem) in terms of *y*." In this interaction between social feedback and self-talk, the individual will identify diverse self-aspects, be it behaviors, physical characteristics, attitudes, values, opinions, beliefs, motivations, and so on.

Now—and this is my point—it can be suggested that *this inter-personal mode of acquisition of self-information* (taking others' perspectives) *could be reproduced by self-talk*, thus leading to self-awareness. We sometimes engage in a fictive dialogue with ourselves in which we state to imaginary persons our motives for behaving in a given way or for having some personal characteristics. When, in response to the imagined speeches of others, we explain our actions or describe ourselves in self-talk, we take others' perspectives into account and thus gain a relatively objective vision of ourselves. Children spontaneously engage in such fictive dialogues. In the following example, David is playing alone with a toy while his private speech is recorded:

> The wheels go here, the wheels go here. Oh, we need to start it all over again. We need to close it up. See, it closes up. We're starting it all over again. *Do you know why we wanted to do that? Because I needed it to go a different way.* Isn't it going to be pretty clever, don't you think? But we have to cover up the motor just like a real car. (Kohlberg, Yaeger, and Hjertholm, 1968, p. 695; italics added)

The first two self-statements in italics represent a good example of self-conversation through which a question about one's action is directed toward a group of imaginary people [of which David seems to be a part, since he says "we"] and is answered by the person emitting the target-behavior [David, the "I"]. Here, an objective vision of oneself and an acquisition of self-information are reached because (a) when David asks to himself "Do you know why we wanted to do that?" he gets to know how other persons could interpret his behavior, and (b) when he answers the question for himself ["Because I needed it to go a different way"], verbal self-information is acquired. David thus reproduces, via self-talk, a process of acquisition of self-information initially triggered by the presence of others.

This process also takes place in adults, but is likely to differ in some respects. First, to my knowledge, adults rarely use the pronoun "we" in their internal dialogue: they fully distinguish themselves from others by using "I" and "you." Moreover, any reference to others in the internal dialogue tends to become highly understated—it even disappears altogether in most instances. An example of such self-talk could be: "I hope my new look will be appreciated!" [objective vision of oneself produced by an anticipation of the reaction of an imaginary group of persons—the *generalized other* of Mead]; "Some might say it gives me a more 'serious' look—that's what I wanted—up to a certain extent, anyway. Obviously, I look older—more 'mature'" [acquisition of self-information]. Another example, in which there is this time an explicit reference to others, would be: "*X* might wonder why I did that. She [or he] should be aware that my relation with *y* is serious, and that although I show her [or him—i.e., *x*] affection, it is *y* that I am in love with. *X* shouldn't feel hurt if I didn't accept her [or his] advances—or maybe I wasn't clear enough to start with?" We can clearly see here in what way perspective taking allows an objective vision of oneself, and in what way the self-observation brought by this objective vision allows in turn the identification of a precise self-aspect—the acquisition of self-information (here: the actions of the person toward *x* and his or her feelings toward *y*). Finally—and more often than otherwise, I would think—any reference to others disappears despite the fact that this reference still motivates the speech for oneself: "How did I look when I gave this lecture? I probably looked nervous—God I *was* nervous! Fortunately, I think I looked competent despite my nervousness." So, one first way to understand the nature of a link between self-talk and self-awareness consists in recognizing that an objective vision of oneself, originally dependent upon the presence of others, can be internally reproduced through inner verbal conversations we have with fictive persons.

Reproducing Appraisals Made by Others by Self-talk

Let me now consider Cooley's thesis according to which individuals emit observations about us that enrich our self-concept. Someone might say to me "I called you three times this week and you didn't call me back. I don't find this very polite! I would even say that you don't show much respect!" Such a remark suggests, as far as it applies to me, that one is *inconsiderate*—that one *lacks respectfulness*. Inner speech allows one to consolidate such self-information: "That's true . . . ." Inner speech also allows oneself to *question* such self-information: "That's untrue! The fact that I didn't call back doesn't mean I lack respectfulness: I was out of town!" But more importantly, *self-talk would allow a reproduction for oneself of these appraisals we get from others*. This is where the nature of a link between self-talk and self-awareness becomes clear. People address to themselves many verbal comments—for instance: "My God you get angry easily!"; "Why did you do that?"; "Did you ever notice this tendency you have to take yourself way too seriously?"; "You're very bright!"; "Admit that this is what you are thinking about!"; "You seem to be a very sensitive person!"; "You are sad, aren't you?" and so on. Such comments (observations and inferences about one's thoughts, feelings and behaviors) by others might, upon repetition, *imprint on one's self-talk a propension to address to ourselves such remarks*. A mode of transmission of self-information that was originally inter-personal (verbal comments made by others about ourselves) would gradually become an *intra*-personal mode of transmission of self-information (verbal comments about ourselves that *we address to ourselves*). Examples of such self-statements would be: "My God I get angry easily!"; "Why did I do that?"; "I take myself way too seriously!"; "I'm bright, sensitive, sad," and so on. (Of course, these examples amount to rather simple, short self-conversations; we often engage in much more sophisticated verbal self-analyses. I could illustrate such lengthy self-conversations, but I must here restrict myself.)

Creating a Redundancy of Self-information by Self-talk

I will now put aside any reference to psychosocial considerations and focus on the intrapsychic world of the individual. The following citation will guide my next proposition concerning the nature of a relation between self-talk and self-awareness:

> A subject completely immersed in experience would not be conscious of it [the experience]. It is a platitude that we are indeed unconscious of most of the background noises, pressures, luminosities, odors, and visceral sensations that impinge upon us at any given moment. We are unaware of them not because they are remote but because they are too near. There is no distance between us and them. . . . A person can be conscious of something only if a wedge has been inserted between him and it. . . . In complete immersion in experience there is no sense of ownership. (Johnstone, 1970, p. 106)

The essence of the above quotation boils down to this: an observation is possible *only if* there exists a *distance* (a wedge) between the observer and the observed thing. By the same token, a *self*-observation is possible *only if* there exists a distance between the individual and any potentially observable self-aspect. A clear relation can be established here between the capacity to operate a backward movement on oneself and self-awareness. By definition, self-awareness represents the capacity to become the object of one's own attention. This, again, can be done by taking others' perspectives: the individual "introduces" himself or herself (in imagination) into someone else and observes himself or herself under this new perspective. In doing so, a (mental) distance is created within the self (between the individual and himself or herself).

I already have suggested how self-talk would allow the individual to take the perspective of others; consequently—and although the discussion centered on another aspect of the problem—we already have seen how self-talk could create a distance within the self. I will now propose that self-talk can *also* create a *redundancy* of self-information within the self, and that such a redundancy creates in turn a distance within the self.[1] Thus, a second way to understand the nature of a link between self-talk and self-awareness will be presented.

The term "redundancy" implies that some already given information, or self-information, is brought under a new form (Robert, 1973). To illustrate, let us imagine an individual experiencing a given subjective feeling—for example, an emotion of joy; this emotion represents a potential bit of self-information. The individual talks to himself or herself and says "God! This is fun!" A replication takes place, and with it comes the same self-information under a new form. How does self-talk produce a redundancy of self-information? In what way is the already given self-information brought under a new form? Self-talk *carries* information. The self-information in question here refers to any *already given* self-aspect (in the above example, an emotion) since it is intrinsic to the individual: self-information is, as a matter of fact,

[1] Routtenberg (1980) proposes a psychobiological model of (self-) awareness in which the notion of "redundancy" in the nervous system plays a central role. His model, however, does not have much in common with the present analysis.

what the individual is. This self-information is brought *under a new form* because any self-information conveyed by self-talk presents itself in *words and sentences*—which is clearly different than an emotion, a physical characteristic, or any other given subjective experience.

Now, how is it that a distance is created between the subjective experience of joy (the already given self-information) and its linguistic representation (the self-statement—the new self-information resulting from the replication)? The individual, before the redundancy, was *immersed* in his or her subjective experience. After the redundancy, he or she now has access in his or her perceptive field to a self-information to which he or she did not have access to previously—we thus have here the creation of a distance. The example illustrates the redundancy of a precise subjective experience; but the principle it allows me to expose applies to any possible personal characteristic as well—to any "private" or "public" self-aspect.

To summarize: self-talk, by verbally identifying self-information that is inherent to the individual, brings it under a new form—hence a redundancy. In producing redundancy within the self, self-talk also creates a distance between self-information and the individual (the self). The individual, as a result, can observe—acquire—the self-information. Self-awareness is dependent upon a distance between the individual and himself or herself; a redundancy of self-information creates such a distance; self-talk in turn creates redundancy. Insofar as this reasoning makes sense, the nature of the relation between self-talk and self-awareness can now be understood within a second perspective.[2]

Conclusion

In humans at least, self-awareness is pretty much taken for granted (Gallup, 1987). It is its *underlying mechanism* that keeps puzzling psychologists and philosophers. What is introspection? How do we have access to the con-

[2] In another article (Morin and Everett, 1990b), I discussed this hypothesis by using the following analogy: the brain would be equipped with "internal mirrors" allowing the subjective experience it generates to detach itself from itself. These internal mirrors could consist of cognitive processes capable of reproducing, as a whole or in part and at a given moment, the subjective experience, thus producing redundancy within the self. Self-talk would be one such cognitive process.

Of course, this "mirror" analogy does not tell us much about the nature of the link between self-talk and self-awareness; but it allows me to counter a difficulty raised by Johnstone (1970). According to this author, to become the object of one's own attention represents a logical impossibility: "*The incapacity of consciousness to be its own object is structurally identical with the incapacity of the pointing needle to point to itself. Consciousness of consciousness is a contradiction*" (p. 105). This argument seems to me to be misleading, because a needle can indeed point to itself *if a mirror is disposed at its extremity*. By extension, an individual placed in front of a mirror can contemplate his or her secular image—that is, he or she can become the object of his or her own attention. This analogy applies to the individual's subjective experience as well, which, by self-reflection, could become the object of its own attention.

tent of our subjective experience or to any other self-aspect? Does self-awareness require some sort of "mental eye" or any other mysterious internal device? These represent highly intriguing questions, and no doubt their answers will prove to be extremely complex.

In this article and elsewhere (see Morin, 1992; Morin and Everett, 1990a, 1990b; Morin et al., 1993), I proposed that more often than not, self-awareness is mediated by self-talk. But only to put forward such an hypothesis is theoretically unsatisfactory: one must also try to understand *how* self-talk mediates self-awareness. As Churchland (1983, p. 88) puts it: "What is it about self-consciousness such that it requires linguistic representations, and what is it about language such that it brings about the special capacity for self-consciousness?"[3]

The following propositions about the nature of the relation between self-talk and self-awareness were explored. The social mechanism initiating the taking of others' perspectives, and resulting in an objective vision of oneself, can be reproduced by self-talk; also, self-talk allows a reproduction for oneself of the appraisals we get from others. And finally, self-talk creates a redundancy of self-information within the self, and with it a distance (essential to self-awareness) between self-information and the individual (the self). Although, at this point, further studies are needed to confirm the existence of a link between self-talk and self-awareness, the cogency of the main propositions put forward in this article should be empirically explored as well. One possibility, for example, might consist in training one group of subjects to have inner verbal conversations about themselves with fictive persons. If self-talk directed toward fictive individuals facilitates perspective taking and increases self-awareness, significant differences in self-awareness should be observed between control and experimental groups. Such attempts to put to test the aforementioned hypotheses about the nature of the relation between self-talk and self-awareness represent one of many possible avenues toward a better understanding of the mechanisms the self uses in thinking of itself.

References

Beck, A.T., Brown, G., Steer, R.A., Eidelson, J.I., and Riskind, J.H. (1987). Differentiating anxiety and depression utilizing the Cognition Checklist. *Journal of Abnormal Psychology*, 96, 179–183.

Bibler, V.S. (1983). Thinking as creation: Introduction to the logic of mental dialogue. *Soviet Psychology*, *22*, 33–54.

Buss, A.H. (1980). *Self-consciousness and social anxiety*. San Francisco: Freeman.

[3] Of course, language does not, alone, "bring about" self-consciousness. Language—and more precisely, self-talk—represents one factor among many others that might contribute to self-awareness. It would be naive to suppose that probably the most complex operation the mind can perform upon itself—to reflect upon itself—could be mediated by a single cognitive process.

Cacioppo, J.T., Glass, C.R., and Merluzzi, T.V. (1979). Self-statements and self-evaluation: A cognitive-response analysis of heterosocial anxiety. *Cognitive Therapy and Research, 3,* 249–262.

Carver, C.S., and Scheier, M.F. (1981). *Attention and self-regulation: A control-theory approach to human behavior.* New York: Springer-Verlag.

Chambless, D.L., Caputo, G.C., Bright, P., and Gallagher, R. (1984). Assessment of fear in agoraphobics: The Body Sensations Questionnaire and the Agoraphobic Cognitions Questionnaire. *Journal of Consulting and Clinical Psychology, 52,* 1090–1097.

Churchland, P.S. (1983). Consciousness: The transmutation of a concept. *Pacific Philosophical Quarterly, 64,* 80–95.

Clark, D.A., and Channon, S. (1987). Differences and dysfunctional thinking between anorexic, bulimic, and student nurse samples. *Cognitive Therapy and Research,* 8, 36–51

Cooley, C.H. (1902). *Human nature and the social order.* New York: Scribners.

Duval, S., and Wicklund, R.A. (1972). *A theory of objective self awareness.* New York: Academic Press.

Fenigstein, A. (1979). Self-consciousness, self-attention, and social interaction. *Journal of Personality and Social Psychology, 37,* 75–86.

Fenigstein, A., Scheier, M.F., and Buss, A.H. (1975). Public and private self-consciousness: Assessment and theory. *Journal of Consulting and Clinical Psychology,* 36, 1241–1250.

Firestone, R.W. (1987). The "voice": The dual nature of guilt reactions. *The American Journal of Psychoanalysis, 47*(3), 210–229.

Froming, N.J., and Carver, C.S. (1981). Divergent influences of private and public self-consciousness in a compliance paradigm. *Journal of Research in Personality, 15,* 159–171.

Fuqua, D.R., Newman, J.L., Anderson, M.W., and Johnson, A.W. (1986). Preliminary study of internal dialogue in a training setting. *Psychological Reports, 58,* 163–172.

Gaboury, A., and Ladouceur, R. (1989). Erroneous perceptions and gambling. *Journal of Social Behavior and Personality,* 4(4), 411–420.

Gallup, G.G., Jr. (1985). Do minds exist in species other than our own? *Neuroscience and Biobehavioral Reviews,* 9, 631–641.

Gallup, G.G., Jr. (1987). Self-awareness. In G. Mitchell and J. Erwin (Eds.), *Comparative primate biology (Vol. 2, Part b): Behavior cognition and motivation* (pp. 3–16). New York: Alan R. Liss, Inc.

Gallup, G.G., Jr., and Suarez, S.D. (1986). Self-awareness and the emergence of mind in humans and other primates. In J. Suls and A. G. Greenwald (Eds.), *Psychological perspectives on the self* (Vol. 3, pp. 3–26). Hillsdale, New Jersey: Erlbaum.

Gibbons, F.X. (1990). Self-attention and behavior: A review and theoretical update. In M.P. Zanna (Ed.), *Advances in experimental social psychology* (Vol. 23, pp. 249–303). New York: Academic Press.

Harris, K.R. (1986). The effects of cognitive-behavior modification on private speech and task performance during problem solving among learning-disabled and normally achieving children. *Journal of Abnormal Child Psychology, 14,* 63–67.

Highlen, P.S., and Bennett, B.B. (1983). Elite divers and wrestlers: A comparison between open and closed skill athletes. *Journal of Sport Psychology,* 5(4), 390–409.

Hunsley, J. (1987). Internal dialogue during academic examinations. *Cognitive Therapy and Research, 11*(6), 653–664.

Ingram, R.E. (1990). Self-focused attention in clinical disorders: Review and a conceptual model. *Psychological Bulletin, 107,* 156–176.

Johnstone, H.W. (1970). *The problem of the self.* University Park, Pennsylvania: Pennsylvania State University Press.

Kline, W.B. (1983). Training counselor trainees to talk to themselves: A method of focusing attention. *Counsellor Education and Supervision, 22,* 296–302.

Kohlberg, L., Yaeger, J., and Hjertholm, E. (1968). Private speech: Four studies and a review of theories. *Child development,* 39, 691–736.

Kurpius, D.J., Benjamin, D., and Morran, D.K. (1985). Effects of teaching a cognitive strategy on counsellor trainee internal dialogue and clinical hypothesis formulation. *Journal of Counseling Psychology,* 32, 263–271.

Ladouceur, R., Gaboury, A., and Duval, C. (1988). Modification des verbalisations irrationnelles pendant le jeu de roulette américaine et prise de risque monétaire [Modification of irrational verbalizations in gambling]. *Science et Comportement, 18*, 58–68.

Luria, A.R. (1959). The directive function of speech in development and dissolution (Part 1): Development of the directive function of speech in early childhood. *Word, 15*(2), 341–352.

Luria, A.R. (1961). *The role of speech in the regulation of normal and abnormal behaviors*. New York: Liveright.

Luria, A.R. (1978). *Les fonctions corticales supérieures de l'homme* [Superior cortical functions in man]. Paris: Presses Univeritaires de France.

MacKay, P. (1979). The game of internalizing others. In J.M. Davidson and R.J. Davidson (Eds.), *The psychobiology of consciousness* (pp. 231–243). New York: Plenum Press.

McCall, G.J. (1977). The social looking-glass: A sociological perspective on self-development. In T. Mischel (Ed.), *The self: Psychological and philosophical issues* (pp. 17–29). Oxford, England: Basil Blackwell.

Meacham, J. A. (1976). The role of verbal activity in remembering the goals of actions. In G. Zivin (Ed.), *The development of self-regulation through private speech* (pp. 237–263). New York: Wiley.

Mead, G.H. (1934). *Mind, self, and society*. Chicago: University of Chicago Press.

Mead, G.H. (1964). The mechanism of social consciousness. In A.J. Reck (Ed.), *Selected writings: George Herbert Mead* (pp. 134–149). Chicago: University of Chicago Press. (First published in 1912)

Mead, G.H. (1982). Consciousness, mind, the self, and scientific objects. In D.L. Miller (Ed.), *The individual and the social self* (pp. 176–196) [Unpublished work of George Herbert Mead]. Chicago: University of Chicago Press.

Meichenbaum, D. (1976). Toward a psychocognitive theory of self-regulation. In G.E. Schwartz and D. Shapiro (Eds.), *Consciousness and self-regulation* (Vol. 1, pp. 121–145). New York: Plenum Press.

Meichenbaum, D. (1977). *Cognitive-behavior modification: An integrative approach*. New York: Plenum Press.

Meichenbaum, D., and Butler, L. (1980). Cognitive ethology: Assessing the stream of cognition and emotion. In K. Blankstein, P. Pliner, and J. Polivy (Eds.), *Advances in the study of communication and affect: Assessment and modification of emotional behavior* (Vol. 6). New York: Plenum Press.

Meltzer, B.N. (1991), Mead on imagery: A complement to Count-van Manen's views. *Journal of Mental Imagery, 15*, 17–33.

Morin, A. (1988). *Langage intérieur, conscience de soi, et auto-régulation: Un point de vue neuropsychologique* [Inner speech, self-consciousness, and self-regulation: A neuropsychological perspective]. Unpublished manuscript, École de Psychologie, Université Laval, Québec.

Morin, A. (1992). *Une exploration théorique et empirique de l'existence d'une relation entre le dialogue intérieur et la conscience de soi* [A theoretical and empirical exploration of the existence of a relation between self-talk and self-awareness]. Unpublished doctoral dissertation. École de Psychologie, Université Laval, Québec.

Morin, A., and DeBlois, S. (1989). Gallup's mirrors: More than an operationalization of self-awareness in primates? *Psychological Reports, 65*, 287–291.

Morin, A., and Everett, J. (1990a). Inner speech as a mediator of self-awareness, self-consciousness, self-knowledge: An hypothesis. *New Ideas in Psychology, 8*, 337–356.

Morin, A., and Everett, J. (1990b). Conscience de soi et langage intérieur: Quelques spéculations [Self-awareness and inner speech: Some speculations]. *Philosophiques, XVII*(2), 169–188.

Morin, A., Everett, J., Turcotte, I., and Tardif, G. (1993). Le dialogue intérieur comme médiateur cognitif de la conscience de soi privée: Une mesure de l'activité de se parler à soi-même à propos de soi et une étude corrélationnelle [Self-talk as a mediator of private self-consciousness: A measure of self-talk and a correlational study]. *Revue Québécoise de Psychologie, 14*(2), 3–19.

Natsoulas, T. (1985). George Herbert Mead's conception of consciousness. *Journal for the Theory of Social Behaviour, 15*, 60–76.

Oei, T.P., and Young, R.M. (1987). The roles of alcohol-related self-statements in social drinking. *International Journal of The Addiction*, *22*(10), 905–915.

Pinard, A. (1992). Métaconscience et métacognition [Metaconsciousness and metacognition]. *Canadian Psychology/Psychologie Canadienne*, *33*, 27–39.

Rimé, B., and LeBon, C. (1984). Le concept de conscience de soi et ses opérationnalisations [The concept of self-consciousness and its operationalizations]. *L'Année Psychologique*, *84*, 535–555.

Robert, P. (1973). *Le Petit Robert: Dictionnaire alphabétique et analogique de la langue française* [Le Petit Robert: French alphabetic and analogic dictionary]. Paris: Société du Nouveau Littré.

Roberts, R.N. (1979). Private speech in academic problem-solving: A naturalistic perspective. In G. Zivin (Ed.), *The development of self-regulation through private speech* (pp. 295–323). New York: Wiley.

Routtenberg, A. (1980). Redundancy in the central nervous system. In J.M. Davidson and R.J. Davidson (Eds.), *The psychobiology of consciousness* (pp. 105–127). New York: Plenum Press.

Scheier, M.F., and Carver, C.S. (1977). Self-directed attention and the experience of emotion: Attraction, repulsion, elation, and depression. *Journal of Personality and Social Psychology*, *35*, 625–636.

Scheier, M.F., and Carver, C.S. (1988). A model of behavioral self-regulation: Translating intention into action. In L. Berkowitz (Ed.), *Advances in experimental social psychology* (Vol. 21, pp. 303–346). New York: Academic Press.

Schwartz, R.M. (1986). The internal dialogue: On the asymmetry between positive and negative coping thoughts. *Cognitive Therapy and Research*, *10*, 591–605.

Sokolov, A.N. (1972). *Inner speech and thought*. New York: Plenum Press.

Turner, R.G. (1976). *Private self-consciousness as a moderator of length of self-description*. Unpublished manuscript, University of Pepperdine.

Uhlemann, M.R., Lee, D.Y., and Hiebert, B. (1988). Self-talk of counsellor trainees: A preliminary report. *Canadian Journal of Counseling / Revue Canadienne de Counseling*, *22*, 73–79.

Yaden, D.B. (1984). Inner speech, oral language, and reading: Huey and Vygotsky revisited. *Reading Psychology*, *5*, 155–166.

Vygotsky, L.S. (1962). *Thought and language*. Cambridge, Massachusetts: MIT Press. (originally published 1934)

Zivin, G. (Ed.). (1979). *The development of self-regulation through private speech*. New York: Wiley.

Identifying Expressions of Emotion in Text

Conference Paper · September 2007

DOI: 10.1007/978-3-540-74628-7_27 · Source: DBLP

Stan Szpakowicz
University of Ottawa

Identifying Expressions of Emotion in Text

Saima Aman[1] and Stan Szpakowicz[1,2]

[1] School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada
[2] Institute of Computer Science, Polish Academy of Sciences, Warszawa, Poland
{saman071, szpak}@site.uottawa.ca

Abstract. Finding emotions in text is an area of research with wide-ranging applications. We describe an emotion annotation task of identifying emotion category, emotion intensity and the words/phrases that indicate emotion in text. We introduce the annotation scheme and present results of an annotation agreement study on a corpus of blog posts. The average inter-annotator agreement on labeling a sentence as emotion or non-emotion was 0.76. The agreement on emotion categories was in the range 0.6 to 0.79; for emotion indicators, it was 0.66. Preliminary results of emotion classification experiments show the accuracy of 73.89%, significantly above the baseline.

1 Introduction

Analysis of sentiment in text can help determine the opinions and affective intent of writers, as well as their attitudes, evaluations and inclinations with respect to various topics. Previous work in sentiment analysis has been done on a variety of text genres, including product and movie reviews [9, 18], news stories, editorials and opinion articles [20], and more recently, blogs [7].

Work on sentiment analysis has typically focused on recognizing valence – positive or negative orientation. Among the less explored sentiment areas is the recognition of types of emotions and their strength or intensity. In this work, we address the task of identifying expressions of emotion in text. Emotion research has recently attracted increased attention of the NLP community – it is one of the tasks at Semeval-2007[1]; a workshop on emotional corpora was also held at LREC-2006[2].

We discuss the methodology and results of an emotion annotation task. Our goal is to investigate the expression of emotion in language through a corpus annotation study and to prepare (and place in the public domain) an annotated corpus for use in automatic emotion analysis experiments. We also explore computational techniques for emotion classification. In our experiments, we use a knowledge-based approach for automatically classifying emotional and non-emotional sentences. The results of the initial experiments show an improved performance over baseline accuracy.

The data in our experiments come from blogs. We wanted emotion-rich data, so that there would be ample examples of emotion use for analysis. Such data is

[1] http://nlp.cs.swarthmore.edu/semeval/tasks/task14/summary.shtml

[2] http://www.lrec-conf.org/lrec2006/IMG/pdf/programWSemotion-LREC2006-last1.pdf

V. Matousek and P. Mautner (Eds.): TSD 2007, LNAI 4629, pp. 196–205, 2007.

expected in personal texts, such as diaries, email, blogs and transcribed speech, and in narrative texts such as fiction. Another consideration in selecting blog text was that such text does not conform to the style of any particular genre *per se*, thus offering a variety in writing styles, choice and arrangement of words, and topics.

2 Related Work

Some researchers have studied emotion in a wider framework of *private states* [12]. Wiebe et al. [20] worked on the manual annotation of private states including emotions, opinions, and sentiment in a 10,000-sentence corpus (the MPQA corpus) of news articles. Expressions of emotions in text have also been studied within the *Appraisal Framework* [5], a functional theory of the language used for conveying attitudes, judgments and emotions [15, 19]. Neither of these frameworks deals exclusively with emotion, the focus of this paper.

In a work focused on learning specific emotions from text, Alm et al. [1] have explored automatic classification of sentences in children's fairy tales according to the basic emotions identified by Ekman [3]. The data used in their experiments was manually annotated with emotion information, and is targeted for use in a text-to-speech synthesis system for expressive rendering of stories. Read [14] has used a corpus of short stories, manually annotated with sentiment tags, in automatic emotion-based classification of sentences. These projects focus on the genre of fiction, with only sentence-level emotion annotations; they do not identify emotion indicators within a sentence, as we do in our work.

In other related work, Liu et al. [4] have utilized real-world knowledge about affect drawn from a common-sense knowledge base. They aim to understand the semantics of text to identify emotions at the sentence level. They begin with extracting from the knowledge base those sentences that contain some affective information. This information is utilized in building affective models of text, which are used to label each sentence with a six-tuple that corresponds to Ekman's six basic emotions [3]. Neviarouskaya et al. [8] have also used a rule-based method for determining Ekman’s basic emotions in the sentences in blog posts.

Mihalcea and Liu [6] have focused in their work on two particular emotions – *happiness* and *sadness*. They work on blog posts which are self-annotated by the blog writers with *happy* and *sad* mood labels. Our work differs in the aim and scope from those projects: we have prepared a corpus annotated with rich emotion information that can be further used in a variety of automatic emotion analysis experiments.

3 The Emotion Annotation Task

We worked with blog posts we collected directly from the Web. First, we prepared a list of seed words for six basic emotion categories proposed by Ekman [3]. These categories represent the distinctly identifiable facial expressions of emotion – *happiness*, *sadness*, *anger*, *disgust*, *surprise* and *fear*. We took words commonly used in the context of a particular emotion. Thus, we chose “happy”, “enjoy”, “pleased” as seed words for the *happiness* category, “afraid”, “scared”, “panic” for the *fear* category, and so on. Next, using the seed words for each category, we retrieved blog posts containing one or more of those words. Table 1 gives the details of the datasets thus collected. Sample examples of annotated text appear in Table 2.

Table 1. The details of the datasets

Dataset	**# posts**	**# sentences**	**Collected using seed words for**
Ec-hp	34	848	*Happiness*
Ec-sd	30	884	*Sadness*
Ec-ag	26	883	*Anger*
Ec-dg	21	882	*Disgust*
Ec-sp	31	847	*Surprise*
Ec-fr	31	861	*Fear*
Total	173	5205	

Table 2. Sample examples from the annotated text

I have to look at life in her perspective, and it would break anyone’s heart. (*sadness, high*)
We stayed in a tiny mountain village called Droushia, and these people brought hospitality to incredible new heights. (*surprise, medium*)
But the rest of it came across as a really angry, drunken rant. (*anger, high*)
And I reallllly want to go to Germany – dang terrorists are making flying overseas all scary and annoying and expensive though!! (*mixed emotion, high*)
I hate it when certain people always seem to be better at me in everything they do. (*disgust, low*)
Which, to be honest, was making Brad slightly nervous. (*fear, low*)

Emotion labeling is reliable if there is more than one judgment for each label. Four judges manually annotated the corpus; each sentence was subject to two judgments. The first author of this paper produced one set of annotations, while the second set was shared by the three other judges. The annotators received no training, though they were given samples of annotated sentences to illustrate the kind of annotations required. The annotated data was prepared over a period of three months.

The annotators were required to label each sentence with the appropriate emotion category, which describes its affective content. To Ekman's six emotions [3], we added *mixed emotion* and *no emotion*, resulting in eight categories to which a sentence could be assigned. While sentiment analysis usually focuses on documents, this work’s focus is on the sentence-level analysis. The main consideration behind this decision is that there is often a dynamic progression of emotions in the narrative texts found in fiction, as well as in the conversation texts and blogs.

The initial annotation effort suggested that in many instances a sentence was found to exhibit more than one emotion – consider (1), for example, marked for both *happiness* and *surprise*. Similarly, (2) shows how more than one type of emotion can be present in a sentence that refers to the emotional states of more than one person.

(1) Everything from trying to order a baguette in the morning to asking directions or talking to cabbies, we were always pleasantly surprised at how open and welcoming they were.

(2) I felt bored and wanted to leave at intermission, but my wife was really enjoying it, so we stayed.

We also found that the emotion conveyed in some sentences could not be attributed to any basic category, for example in (3). We decided to have an additional category called *mixed emotion* to account for all such instances. All sentences that had no emotion content were to be assigned to the *no emotion* category.

(3) It's like everything everywhere is going crazy, so we don't go out any more.

In the final annotated corpus, the *no emotion* category was the most frequent. It is important to have *no emotion* sentences in the corpus, as both *positive* and *negative* examples are required to train any automatic analysis system. It should also be noted that in both sets of annotations a significant number of sentences were assigned to the *mixed emotion* category, justifying its addition in the first place.

The second kind of annotations involved assigning emotion intensity (*high, medium*, or *low*) to all emotion sentences in the corpus, irrespective the emotion category assigned to them. No intensity label was assigned to the *no emotion* sentences. A study of emotion intensity can help recognize the linguistic choices writers make to modify the strength of their expressions of emotion. The knowledge of emotion intensity can also help locate highly emotional snippets of text, which can be further analyzed to identify emotional topics. Intensity values can also help distinguish borderline cases from clear cases [20], as the latter will generally have higher intensity.

Besides labeling the emotion category and intensity, the secondary objective of the annotation task was to identify spans of text (individual words or strings of consecutive words) that convey emotional content in a sentence. We call them emotion indicators. Knowing them could help identify a broad range of affect-bearing lexical tokens and possibly, syntactic phrases. The annotators were permitted to mark in a sentence any number of emotion indicators of any length.

We considered several annotation schemes for emotion indicators. First we thought to identify only individual words for this purpose. That would simplify calculating the agreement between annotation sets. We soon realized, however, that individual words may not be sufficient. Emotion is often conveyed by longer units of text or by phrases, for example, the expressions “can't believe” and “blissfully unaware” in (4). It would also allow the study of the various linguistic features that serve to emphasize or modify emotion, as the use of word “blissfully” in (4) and “little” in (5).

(4) I can't believe this went on for so long, and we were blissfully unaware of it.

(5) The news brought them little happiness.

4 Measuring Annotation Agreement

The interpretation of sentiment information in text is highly subjective, which leads to disparity in the annotations by different judges. Difference in skills and focus of the judges, and ambiguity in the annotation guidelines and in the annotation task itself also contribute to disagreement between the judges [11]. We seek to find how much the judges agree in assigning a particular annotation by using metrics that quantify these agreements.

First we measure how much the annotators agree on classifying a sentence as an emotion sentence. Cohen's kappa [2] is popularly used to compare the extent of consensus between judges in classifying items into known mutually exclusive categories. Table 3 shows the pair-wise agreement between the annotators on emotion/non-emotion labeling of the sentences in the corpus. We report agreement values for pairs of annotators who worked on the same portion of the corpus.

Table 3. Pair-wise agreement in emotion/non-emotion labeling

	a↔b	**a↔c**	**a↔d**	**average**
Kappa	0.73	0.84	0.71	0.76

Table 4. Pair-wise agreement in emotion categories

Category	**a↔b**	**a↔c**	**a↔d**	**average**
happiness	0.76	0.84	0.71	0.77
sadness	0.68	0.79	0.56	0.68
anger	0.62	0.76	0.59	0.66
disgust	0.64	0.62	0.74	0.67
surprise	0.61	0.72	0.48	0.60
fear	0.78	0.80	0.78	0.79
mixed emotion	0.24	0.61	0.44	0.43

Within the emotion sentences, there are seven possible categories of emotion to which a sentence can be assigned. Table 4 shows the value of kappa for each of these emotion categories for each annotator pair. The agreement was found to be highest for *fear* and *happiness*. From this, we can surmise that writers express these emotions in more explicit and unambiguous terms, which makes them easy to identify. The *mixed emotion* category showed least agreement which was expected, given the fact that this category was added to account for the sentences which had more than one emotions, or which would not fit into any of the six basic emotion categories.

Agreement on emotion intensities can also be measured using kappa, as there are distinct categories – *high, medium,* and *low.* Table 5 shows the values of inter-annotator agreement in terms of kappa for each emotion intensity. The judges agreed more when the emotion intensity was high; agreement declined with decrease in the intensity of emotion. It is a major factor in disagreement that where one judge perceives a low-intensity, another judge may find no emotion.

Table 5. Pair-wise agreement in emotion intensities

Intensity	a↔b	a↔c	a↔d	average
High	0.69	0.82	0.65	0.72
Medium	0.39	0.61	0.38	0.46
Low	0.31	0.50	0.29	0.37

Emotion indicators are words or strings of words selected by annotators as marking emotion in a sentence. Since there are no predefined categories in this case, we cannot use kappa to calculate the agreement between judges. Here we need to find agreement between the sets of text spans selected by the two judges for each sentence.

Several methods of measuring agreement between sets have been proposed. For our task, we chose the measure of agreement on set-valued items (MASI), previously used for measuring agreement on co-reference annotation [10] and in the evaluation of automatic summarization [11]. MASI is a distance between sets whose value is 1 for identical sets, and 0 for disjoint sets. For sets A and B it is defined as:

MASI = J * M, where the Jaccard metric is

$$J = |A \cap B| \,/\, |A \cup B|$$

and monotonicity is

$$M = \begin{cases} 1, \mathit{if}\ A = B \\ 2/3, \mathit{if}\ A \subset B\, or\, B \subset A \\ 1/3, \mathit{if}\ A \cap B \neq \phi,\ A - B \neq \phi,\ and\ B - A \neq \phi \\ 0, \mathit{if}\ A \cap B = \phi \end{cases}$$

If one set is monotonic with respect to another, one set's elements always match those of the other set – for instance, in annotation sets {crappy} and {crappy, best} for (6). However, in non-monotonic sets, as in {crappy, relationship} and {crappy, best}, there are elements not contained in one or the other set, indicating a greater degree of disagreement. The presence of monotonicity factor in MASI therefore ensures that the latter cases are penalized more heavily than the former.

While looking for emotion indicators in a sentence, often it is likely that the judges may identify the same expression but differ in marking text span boundaries. For example in sentence (6) the emotion indicator identified by two annotators are “crappy” and “crappy relationship”, which essentially refer to the same item, but disagree on the placement of the span boundary. This leads to strings of varying lengths. To simplify the agreement measurement, we split all strings into words to ensure that members of the set are all individual words. MASI was calculated for each pair of annotations for all sentences in the corpus (see Table 6).

(6) We've both had our share of crappy relationship, and are now trying to be the best we can for each other.

We adopted yet another method of measuring agreement between emotion indicators. It is a variant of the IOB encoding [13] used in text chunking and named entity recognition tasks. We use IO encoding, in which each word in the sentence is labeled as being either In or Outside an emotion indicator text span, as shown in (7).

(7) Sorry/I for/O the/O ranting/I post/O, but/O I/O am/O just/O really/I annoyed/I.

Binary IO labeling of each word in essence reduces the task to that of word-level classification into non-emotion and emotion indicator categories. It follows that kappa can now be used for measuring agreement; pair-wise kappa values using this method are shown in Table 6. The average kappa value of 0.66 is lower than that observed at sentence level classification. This is in line with the common observation that agreement on lower levels of granularity is generally found to be lower.

Table 6. Pair-wise agreement in emotion indicators

Metric	**a↔b**	**a↔c**	**a↔d**	**average**
MASI	0.59	0.66	0.59	0.61
Kappa	0.61	0.73	0.65	0.66

5 Automatic Emotion Classification

Our long-term research goal is fine-grained automatic classification of sentences on the basis of emotion categories. The initial focus is on recognizing emotional sentences in text, regardless of their emotion category. For this experiment, we extracted all those sentences from the corpus for which there was consensus among the judges on their emotion category. This was done to form a gold standard of emotion-labeled sentences for training and evaluation of classifiers. Next, we assigned all emotion category sentences to the class "EM", while all no emotion sentences were assigned to the class "NE". The resulting dataset had 1466 sentences belonging to the EM class and 2800 sentences belonging to the NE class.

5.1 Feature Set

In defining the feature set for automatic classification of emotional sentences, we were looking for features which distinctly characterize emotional expressions, but are not likely to be found in the non-emotional ones. The most appropriate features that distinguish emotional and non-emotional expressions are obvious emotion words present in the sentence. To recognize such words, we used two publicly available lexical resources – the General Inquirer [16] and WordNet-Affect [17].

The General Inquirer (GI) is a useful resource for content analysis of text. It consists of words drawn from several dictionaries and grouped into various semantic categories. It lists different senses of a term and for each sense it provides several tags indicating the different semantic categories it belongs to. We were interested in the tags representing emotion-related semantic categories. The tags we found relevant are *EMOT* (emotion) – used with obvious emotion words; *Pos/Pstv* (positive) and *Neg/Ngtv* (negative) – used to indicate the valence of emotion-related words; *Intrj* (interjections); and *Pleasure* and *Pain*.

WordNet-Affect (WNA) assigns a variety of affect labels to a subset of synsets in WordNet. We utilized the publicly available lists[3] extracted from WNA, consisting of emotion-related words. There are six lists corresponding to the six basic emotion categories identified by Ekman [3].

Beyond emotion-related lexical features, we note that the emotion information in text is also expressed through the use of symbols such as emoticons and punctuation (such as "!"). We, therefore, introduced two more features to account for such symbols. All features are summarized in Table 7 (the feature vector represented counts for all features).

Table 7. Features Used in emotion classification

GI Features	WN-Affect Features	Other Features
Emotion words Positive words Negative words Interjection words Pleasure words Pain words	Happiness words Sadness words Anger words Disgust words Surprise words Fear words	Emoticons Exclamation ("!") and question ("?") marks

5.2 Experiments and Results

For our binary classification experiments, we used Naïve Bayes, and Support Vector Machines (SVM), which have been popularly used in sentiment classification tasks [6, 9]. All experiments were performed using stratified ten-fold cross validation. The naïve baseline for our experiments was 65.6%, which represents the accuracy achieved by assigning the label of the most frequent class (which in our case is NE) to all the instances in the dataset. Each sentence was represented by a 14-value vector, representing the number of occurrences of each feature type in the sentence. Table 9 shows the classification accuracy obtained with the Naïve Bayes and SVM text classifiers. The highest accuracy achieved was 73.89% using SVM, which is higher than the baseline. The improvement is statistically significant (we used the paired t-test, p=0.05).

To explore the contribution of different feature groups to the classification performance, we conducted experiments using (1) features from GI only, (2) features from WordNet-Affect only, (3) combined features from GI and WordNet-Affect, and (4) all features (including the non-lexical features). We achieved the best results when

Table 8. Emotion classification accuracy

Features	**Naïve Bayes**	**SVM**
GI	71.45%	71.33%
WN-Affect	70.16%	70.58%
GI+WN-Affect	71.7%	73.89%
ALL	**72.08%**	**73.89%**

[3] http://www.cse.unt.edu/~rada/affectivetext/data/WordNetAffectEmotionLists.tar.gz

all the features were combined. While the use of non-lexical features does not seem to affect results of SVM, it did increase the accuracy of the Naïve Bayes classifier. This suggests that a combination of features is needed to improve emotion classification results.

The results of the automatic emotion classification experiments show how external knowledge resources can be leveraged in identifying emotion-related words in text. We note, however, that lexical coverage of these resources may be limited, given the informal nature of online discourse. For instance, one of the most frequent words used for *happiness* in the corpus is the acronym "lol", which does not appear in any of these resources. In future experiments, we plan to augment the word lists obtained from GI and WordNet-Affect with such words. Furthermore, in our experiments, we have not addressed the case of typographical errors and orthographic features (for e.g. "soo sweeet") that express or emphasize emotion in text.

We also note that the use of emotion-related words is not the sole means of expressing emotion. Often a sentence, which otherwise may not have an emotional word, may become emotion-bearing depending on the context or underlying semantic meaning. Consider (8), for instance, which implicitly expresses *fear* without the use of any emotion bearing word.

(8) What if nothing goes as planned?

Therefore to be able to accurately classify emotion, we need to do contextual and semantic analysis as well.

6 Conclusion and Future Work

We address the problem of identifying expressions of emotion in text. We describe the task of annotating sentences in a blog corpus with information about emotion category and intensity, as well as emotion indicators. An annotation agreement study shows variation in agreement among judges for different emotion categories and intensity. We found the annotators to agree most in identifying instances of fear and happiness. We found that agreement on sentences with high emotion intensity surpassed that on the sentences with medium and low intensity. Finding emotion indicators in a sentence was found to be a hard task, with judges disagreeing in identifying precisely the spans of text that indicate emotion in a sentence.

We also present the results of automatic emotion classification experiments, which utilized knowledge resources in identifying emotion-bearing words in sentences. The accuracy is 73.89%, significantly higher than our baseline accuracy.

This paper described the first part of an ongoing work on the computational analysis of expressions of emotions in text. In our future work, we will use the annotated data for fine-grained classification of sentences on the basis of emotion categories and intensity. As discussed before, we plan to incorporate methods for addressing the special needs of the kind of language used in online communication. We also plan on using a corpus-driven approach in building a lexicon of emotion words. In this direction, we intend to start with the set of emotion indicators identified during the annotation process, and further extend that using similarity measures.

References

1. Alm, C.O., Roth, D., Sproat, R.: Emotions from text: machine learning for text-based emotion prediction. In: Proc. of the Joint Conf. on Human Language Technology/Empirical Methods in Natural Language Processing (HLT/EMNLP), pp. 579–586 (2005)
2. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20, 37–46 (1960)
3. Ekman, P.: An Argument for Basic Emotions. Cognition and Emotion. 6, 169–200 (1992)
4. Liu, H., Lieberman, H., Selker, T.: A Model of Textual Affect Sensing using Real-World Knowledge. In: Proc. of the Int'l Conf. on Intelligent User Interfaces (2003)
5. Martin, J.R., White, P.R.R.: The Language of Evaluation: Appraisal in English, Palgrave, London (2005), http://grammatics.com/appraisal/
6. Mihalcea, R., Liu, H.: A corpus-based approach to finding happiness. In: The AAAI Spring Symposium on Computational Approaches to Weblogs, Stanford, CA (2006)
7. Mishne, G., Glance, N.: Predicting Movie Sales from Blogger Sentiment. In: AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs (2006)
8. Neviarouskaya, A., Prendinger, H., Ishizuka, M.: Analysis of affect expressed through the evolving language of online communication. In: Proc. of the 12th Int'l Conf. on Intelligent User Interfaces (IUI-07), Honolulu, Hawaii, pp. 278–281 (2007)
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proc. Conf. on EMNLP (2002)
10. Passonneau, R.: Computing reliability for coreference annotation. In: Proc. International Conf. on Language Resources and Evaluation, Lisbon (2004)
11. Passonneau, R.J.: Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. In: Proc. 5th Int'l Conf. on Language Resources and Evaluation (2006)
12. Quirk, R., Greenbaum, S., Leech, G., Svartvik, J.: A Comprehensive Grammar of the English Language. Longman, New York (1985)
13. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Proc. Third ACL Workshop on Very Large Corpora (1995)
14. Read, J.: Recognising affect in text using pointwise mutual information. Master's thesis, University of Sussex (2004)
15. Read, J., Hope, D., Carroll, J.: Annotating expressions of Appraisal in English. In: The Proc. of the ACL Linguistic Annotation,Workshop, Prague (2007)
16. Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilvie, D.M., et al.: The General Inquirer: A Computer Approach to Content Analysis. The MIT Press, Cambridge (1966)
17. Strapparava, C., Valitutti, A.: WordNet-Affect: an affective extension of WordNet. In: Proc. 4th International Conf. on Language Resources and Evaluation, Lisbon (2004)
18. Turney, P.D.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: Proc. the 40th Annual Meeting of the ACL, Philadelphia (2002)
19. Whitelaw, C., Garg, N., Argamon, S.: Using Appraisal Taxonomies for Sentiment Analysis. In: Proc. of the 2nd Midwest Comp., Linguistic Colloquium, Columbus (2005)
20. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation 39(2-3), 165–210 (2005)

Joint Sentiment/Topic Model for Sentiment Analysis

Chenghua Lin
School of Engineering, Computing and Mathematics
University of Exeter
North Park Road, Exeter EX4 4QF, UK
cl322@exeter.ac.uk

Yulan He
Knowledge Media Institute
The Open University
Milton Keynes MK7 6AA, UK
y.l.he.01@cantab.net

ABSTRACT

Sentiment analysis or opinion mining aims to use automated tools to detect subjective information such as opinions, attitudes, and feelings expressed in text. This paper proposes a novel probabilistic modeling framework based on Latent Dirichlet Allocation (LDA), called joint sentiment/topic model (JST), which detects sentiment and topic simultaneously from text. Unlike other machine learning approaches to sentiment classification which often require labeled corpora for classifier training, the proposed JST model is fully unsupervised. The model has been evaluated on the movie review dataset to classify the review sentiment polarity and minimum prior information have also been explored to further improve the sentiment classification accuracy. Preliminary experiments have shown promising results achieved by JST.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms

Algorithms, Experimentation

Keywords

Sentiment analysis, Opinion mining, Latent Dirichlet Allocation, Joint sentiment/topic model

1. INTRODUCTION

As propelled by the rapid growth of text data, text mining has been applied to discover hidden knowledge from text in many applications and domains. In business sectors, great efforts have been made to find out customers' sentiments and opinions, often expressed in free text, towards companies' products and services. However, discovering sentiments and opinions through manual analysis of a large volume of textual data is extremely difficult. Hence, in recent years, there have been much interests in the natural language processing community to develop novel text mining techniques with the capability of accurately extracting customers' opinions from large volumes of unstructured text data.

Among various opinion mining tasks, one of them is sentiment classification, i.e. whether the semantic orientation of a text is positive, negative or neutral. When applying machine learning to sentiment classification, most existing approaches rely on supervised learning models trained from labeled corpora where each document has been labeled as positive or negative prior to training. Such labeled corpora are not always easily obtained in practical applications. Also, sentiment classification models trained on one domain might not work at all when moving to another domain. Furthermore, in a more fine-grained sentiment classification problem (e.g. finding users' opinions for a particular product feature), topic/feature detection and sentiment classification are often performed in a two-stage pipeline process, by first detecting a topic/feature and later assigning a sentiment label to that particular topic.

Intuitively, sentiment polarities are dependent on topics or domains. Therefore, detecting both topic and sentiment simultaneously should serve a critical function in helping users in terms of opinion mining and summarization. For instance, though the adjective 'unpredictable' in a phrase such as 'unpredictable steering' may have negative orientation in an automobile review, it could also have positive orientation in a phrase like 'unpredictable plot' in a movie review [5].

Although much work has been done in detecting topics [2, 6, 20], these lines of work mainly focused on discovering and analyzing topics of documents alone, without any analysis of sentiment in the text, which limit the usefulness of the mining results. Other work [16, 22, 11, 15, 4, 3, 25] addressed the problem of sentiment detection in various levels (i.e. from word/phrase level, to sentence and document level). However, none of them can model mixture of topics alongside with sentiment classification, which again makes the results less informative to users. Some of the recent work [14, 19] has been aware of this limitation and tried to capture sentiments and mixture of topics simultaneously. However, Mei *et al.* [14] does not model sentiment directly and requires post-processing to calculate the positive/negative coverage in a document in order to identify its polarity. Titov and McDonald [19] requires some kind of supervised settings that the customer reviews should contain ratings for the aspects/features discussed in the text and thus it lacks of the flexibility to adapt to other domains.

CIKM'09, November 2–6, 2009, Hong Kong, China.
Copyright 2009 ACM 978-1-60558-512-3/09/11 ...$10.00.

In this paper, we focus on document level sentiment classification based on the proposed unsupervised joint sentiment/topic (JST) model. This model extends the state-of-the-art topic model, Latent Dirichlet Allocation (LDA), by adding a sentiment layer. Our model distinguishes from other models in that: (1) JST is fully unsupervised; (2) JST can detect sentiment and topic simultaneously. To the best of our knowledge, no other existing approaches present the same merits as our model. We have also explored various approaches for obtaining prior information in order to improve the sentiment detection accuracy. Although the proposed JST model can be easily extended to detect polarity of text at various granularity levels, in this paper we mainly focus on reporting our preliminary results on the document-level sentiment classification and briefly present the sentiment analysis results on some extracted topics as an example illustration.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 presents the Joint Sentiment/Topic (JST) model. We show the experimental setup in Section 4 and discuss the results based on the movie review dataset[1] in Section 5. Finally, Section 6 concludes the paper and outlines the future work.

2. RELATED WORK

Great bulk of work has been focused on the problem of sentiment classification at various levels using machine learning techniques. Turney and Littman [22] applied an unsupervised learning algorithm to classify the semantic orientation in the word/phrase level, based on mutual information between document phrases and a small set of positive/negative paradigm words like “good” and “bad”. Choi *et al.* [4] dealt with opinion analysis by combining conditional random fields (CRFs) and a variation of Autoslog. In the sentence level, a semi-supervised machine learning algorithm was proposed by Pang and Lee [15], which employs a subjectivity detector and minimum cuts in graphs. Another system by Kim and Hovy [11] judges the sentiment of a given sentence by combining the individual word-level sentiment. Eguchi and Lavrenko [5] proposed a generative model that jointly models sentiment words, topic words and sentiment polarity in a sentence as a triple. In more recent work [25], the authors tackled this problem utilizing CRFs and considered both contextual dependency and label redundancy in sentence sentiment classification. Another line of work is in the document level, where one tries to evaluate the overall sentiment of a document. The representative work at the early stage can be found in [21, 16], where the former used unsupervised learning and mutual information, which is similar to the approach proposed in [22]; while the latter classified the polarity of movie reviews with the traditional supervised text categorization methods. Following this way, lots of other approaches have been proposed. For example, McDonald *et al.* [13] investigated a global structured model that learns to predict sentiment of different levels of granularity in text. Blitzer *et al.* [3] focused on domain adaption for sentiment classifiers with respect to different types of products’ online reviews.

However, as can be easily pointed out, all the aforementioned work shares some similar limitations: (1) they only focus on sentiment classification without considering the mixture of topics in the text, which is less informative to users and may limit the usefulness of the results; (2) most of the approaches [16, 15, 4, 3, 13, 25] are favored in supervised learning, which require a labeled corpus for training and potentially restrain their applicability to other domains of interest.

Motivated by these observations, we construct an unsupervised hierarchical Bayesian model which can classify document level sentiment and extract mixture of topics simultaneously. To the best of our knowledge, not much work has been done regarding this particular problem. However, there are indeed several lines of work which are quite close to our vision [14, 20, 19].

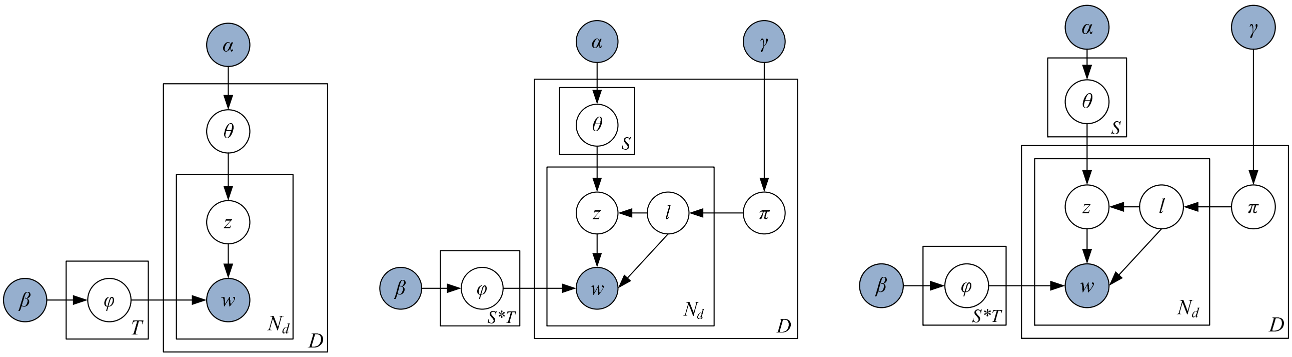
One of the most closely related work is the Topic-Sentiment Model (TSM) [14], which jointly models the mixture of topics and sentiment predictions for the entire document. However, there are several intrinsical differences between JST and TSM. First of all, TSM is essentially based on the Probabilistic Latent Semantic Indexing (pLSI) [8] model with an extra background component and two additional sentiment subtopics, and thus suffers from the problems of inferencing on new document and overfitting the data, both of which are known as the deficits of pLSI. JST overcomes these shortcomings as it is based on LDA with a better statistical foundation. Regarding topic extraction, TSM samples a word either from the background component model or topical themes where the latter are further categorized into three sub-categories, i.e. neutral, positive and negative sentiment models. In contrast, in JST one draws a word from the distribution over words jointly defined by topic and sentiment label that chosen in the first place. Thirdly, for sentiment detection, TSM requires postprocessing to calculate the sentiment coverage of a document, while in JST the document sentiment can be directly obtained from the probability distribution of sentiment label given document.

Other models by Titov and McDonald [20, 19] are also closely related to ours, since they are all based on the state-of-the-art topic model LDA. First proposed in [20], the Multi-Grain Latent Dirichlet Allocation model (MG-LDA) is argued to be more appropriate to build topics that are representative of ratable aspects of objects from online user reviews, by allowing terms being generated from either a global topic or a local topic. Being aware of the limitation that MG-LDA is still purely topic based without considering the associations between topics and sentiments, Titov and McDonald further proposed the Multi-Aspect Sentiment model (MAS) [19] by extending the MG-LDA framework. The major improvement of MAS is that it can aggregate sentiment texts for the sentiment summary of each rating aspect extracted from the MG-LDA. Our model differs from MAS in several aspects: MAS works on a supervised setting as it requires that every aspect is rated at least in some documents, which is practically infeasible in real life applications, while our JST model is fully unsupervised with only minimum prior information being incorporated, which in turn has more flexibilities; MAS focuses on extracting text for sentiment summaries of each aspect ratings while we predict the sentiment orientation in the document level.

3. JOINT SENTIMENT/TOPIC (JST) MODEL

The Latent Dirichlet Allocation (LDA) model, as shown

[1] Polarity dataset v2.0 URL: http://www.cs.cornell.edu/people/pabo/movie-review-data/



(a) (b) (c)

Figure 1: (a) LDA model; (b) JST model; (c) Tying-JST model.

in Figure 1(a), is one of the most popular topic models based upon the assumption that documents are mixture of topics, where a topic is a probability distribution over words [2, 18]. The LDA model is effectively a generative model from which a new document can be generated in a predefined probabilistic procedure. Compared to another commonly used generative model Probabilistic Latent Semantic Indexing (pLSI) [8], LDA has a better statistical foundation by defining the topic-document distribution θ, which allows inferencing on new document based on previously estimated model and avoids the problem of overfitting, where both are known as the deficits of pLSI. Generally, the procedure of generating each word in a document under LDA can be broken down into two stages. One firstly chooses a distribution over a mixture of K topics. Following that, one picks up a topic randomly from the topic distribution, and draws a word from that topic according to the topic's word probability distribution.

The existing framework of LDA has three hierarchical layers, where topics are associated with documents, and words are associated with topics. In order to model document sentiments, we propose a joint sentiment/topic (JST) model by adding an additional sentiment layer between the document and the topic layer. Hence, JST is effectively a four-layer model, where sentiment labels are associated with documents, under which topics are associated with sentiment labels and words are associated with both sentiment labels and topics. A graphical model of JST is represented in Figure 1(b).

Assume that we have a corpus with a collection of D documents denoted by $C = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_D\}$; each document in the corpus is a sequence of N_d words denoted by $\mathbf{d} = (w_1, w_2, ..., w_{N_d})$, and each word in the document is an item from a vocabulary index with V distinct terms denoted by $\{1, 2, ..., V\}$. Also, let S be the number of distinct sentiment labels, and T be the total number of topics. The procedure of generating a word w_i in document d boils down to three stages. Firstly, one chooses a sentiment label l from the document specific sentiment distribution π_d. Following that, one chooses a topic randomly from the topic distribution $\theta_{l,d}$, where $\theta_{l,d}$ is chosen conditioned on the sentiment label l. It is worth noting at this point that the topic-document distribution of JST is different from the one of LDA. In LDA, there is only one topic-document distribution θ for each individual document. In contrast, each document in JST is associated with S (number of sentiment labels) topic-document distributions, each of which corresponds to a sentiment label l with the same number of topics. This feature essentially provides means for the JST model to measure the sentiment of topics. Finally, one draws a word from distribution over words defined by the topic and sentiment label, which is again different from LDA that a word is sampled from the word distribution only defined by topic.

The formal definition of the generative process which corresponds to the hierarchical Bayesian model shown in Figure 1(b) is as follows:

- For each document d, choose a distribution $\pi_d \sim Dir(\gamma)$.
- For each sentiment label l under document d, choose a distribution $\theta_{d,l} \sim Dir(\alpha)$.
- For each word w_i in document d
  - choose a sentiment label $l_i \sim \pi_d$,
  - choose a topic $z_i \sim \theta_{d,l_i}$,
  - choose a word w_i from the distribution over words defined by the topic z_i and sentiment label l_i, $\varphi^{l_i}_{z_i}$.

The hyperparameters α and β in JST can be treated as the prior observation counts for the number of times topic j associated with sentiment label l sampled from a document and the number of times words sampled from topic j associated with sentiment label l respectively, before having observed any actual words. Similarly, the hyperparameter γ can be interpreted as the prior observation counts for the number of times sentiment label l sampled from document before any words from the corpus is observed. In JST, there are three sets of latent variables that we need to infer, including: the joint sentiment/topic-document distribution θ, the joint sentiment/topic-word distribution φ, and the sentiment-document distribution π. We will see later in the paper that the sentiment-document distribution π plays an important role in determining the document polarity.

In order to obtain the distributions of θ, φ and π, we firstly estimate the posterior distribution over z, i.e the assignment of word tokens to topics and sentiment labels. The sampling distribution for a word given the remaining topics and sentiment labels is $P(z_t = j, l_t = k | \mathbf{w}, \mathbf{z_{-t}}, \mathbf{l_{-t}}, \alpha, \beta, \gamma)$ where $\mathbf{z_{-t}}$ and $\mathbf{l_{-t}}$ are vector of assignments of topics and labels for all the words in the collection except for the word at position t in document d.

The joint probability of the topic/sentiment label assignments and the words can be factored into the following three terms:

$$P(\mathbf{w}, \mathbf{z}, \mathbf{l}) = P(\mathbf{w}|\mathbf{z}, \mathbf{l})P(\mathbf{z}, \mathbf{l}) = P(\mathbf{w}|\mathbf{z}, \mathbf{l})P(\mathbf{z}|\mathbf{l}, d)P(\mathbf{l}|d) \tag{1}$$

For the first term, by integrating out φ, we obtain:

$$P(\mathbf{w}|\mathbf{z}, \mathbf{l}) = \left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V}\right)^{T*S} \prod_j \prod_k \frac{\prod_i \Gamma(N_{i,j,k} + \beta)}{\Gamma(N_{j,k} + V\beta)} \tag{2}$$

where V is the size of the vocabulary, T is the total number of topics, S is the total number of sentiment labels, $N_{i,j,k}$ is the number of times word i appeared in topic j and with sentiment label k. $N_{j,k}$ is the number of times words assigned to topic j and sentiment label k, and Γ is the gamma function.

For the second term, by integrating out θ, we obtain:

$$P(\mathbf{z}|\mathbf{l}, d) = \left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\right)^{S*D} \prod_k \prod_d \frac{\prod_j \Gamma(N_{j,k,d} + \alpha)}{\Gamma(N_{k,d} + T\alpha)} \tag{3}$$

where S is the total number of sentiment labels, D is the total number of documents in the collection, $N_{j,k,d}$ is the number of times a word from document d has been associated with topic j and sentiment label k. $N_{k,d}$ is the number of times sentiment label k has been assigned to some word tokens in document d.

For the third term, by integrating out π, we obtain:

$$P(\mathbf{l}|d) = \left(\frac{\Gamma(S\gamma)}{\Gamma(\gamma)^S}\right)^{D} \prod_d \frac{\prod_k \Gamma(N_{k,d} + \gamma)}{\Gamma(N_d + S\gamma)} \tag{4}$$

where D is the total number of documents in the collection, $N_{k,d}$ is the number of times sentiment label k has been assigned to some word tokens in document d. N_d is the total number of words in the document collection.

Gibbs sampling will sequentially sample each variable of interest, z_t and l_t here, from the distribution over that variable given the current values of all other variables and the data. Letting the subscript $-t$ denote a quantity that excludes data from t^{th} position, the conditional posterior for z_t and l_t is:

$$P(z_t = j, l_t = k|\mathbf{w}, \mathbf{z}_{-t}, \mathbf{l}_{-t}, \alpha, \beta, \gamma) \propto$$
$$\frac{\{N_{w_t,j,k}\}_{-t} + \beta}{\{N_{j,k}\}_{-t} + V\beta} \cdot \frac{\{N_{j,k,d}\}_{-t} + \alpha}{\{N_{k,d}\}_{-t} + T\alpha} \cdot \frac{\{N_{k,d}\}_{-t} + \gamma}{\{N_d\}_{-t} + S\gamma} \tag{5}$$

Equation 5 is the conditional probability derived by marginalizing out the random variables φ, θ, and π. A sample obtained from the Markov chain can be used to approximate the distribution of words in topics and sentiment labels:

$$\varphi_{i,j,k} = \frac{N_{i,j,k} + \beta}{N_{j,k} + V\beta} \tag{6}$$

The approximated predictive distribution over topics for sentiment label is:

$$\theta_{j,k,d} = \frac{N_{j,k,d} + \alpha}{N_{k,d} + T\alpha} \tag{7}$$

Finally, the approximated predictive distribution over sentiment label for document is:

$$\pi_{k,d} = \frac{N_{k,d} + \gamma}{N_d + S\gamma} \tag{8}$$

The pseudo code for the Gibbs sampling procedure of JST is shown in Figure 2.

```
1. Initialize $V \times T \times S$ matrix $\mathbf{\Phi}$, $T \times S \times D$ matrix
   $\mathbf{\Theta}$, $S \times D$ matrix $\mathbf{\Pi}$.
2. for $m = 1$ to $M$ Gibbs sampling iterations do
3.     Read a word $i$ from a document
4.     Calculate the probability of assigning word $i$
       to topic and sentiment label based on Equation 5.
5.     Sample a topic $j$ based on the estimated probability obtained in Step 3.
6.     Sample a sentiment label $k$.
7.     Update the matrix $\mathbf{\Phi}$, $\mathbf{\Theta}$, and $\mathbf{\Pi}$ with new
       sampling results.
8.     Go to step 3 until all words have been
       processed.
9. end for
```

Figure 2: Gibbs sampling procedure.

3.1 Tying-JST Model

A variation of JST model is presented in Figure 1(c), namely tying-JST model. The major difference between tying-JST and JST model falls into that, in order to sample a word in a document during the generative process, one has to choose a topic-document distribution θ_d for every document under the JST model, whereas in tying-JST there is only one topic-document distribution θ which accounts for all the documents in the corpus. Therefore, during the Gibbs sampling procedure, rather than having a $\mathbf{\Theta}$ matrix with dimension $T \times S \times D$ as for JST, the $\mathbf{\Theta}$ matrix of tying-JST has only $T \times S$ dimension. As a result, the approximated predictive distribution over topics for sentiment label is different from Equation 7 and should be reformulated as:

$$\theta_{j,k} = \frac{N_{j,k} + \alpha}{N_k + T\alpha} \tag{9}$$

where T is the total number of topics, $N_{j,k}$ is the total number of times topic j is associated with sentiment label k, and N_k is total number of times that a word is associated with sentiment label k.

Experimental results will be presented in Section 5 to compare the performance of the JST and the tying-JST model.

4. EXPERIMENTAL SETUP

In this section, we present the experimental setup of document polarity classification and topic extraction based on the movie review dataset. This dataset consists of two categories of free format movie review texts, with their overall sentiment polarity labeled either positive or negative. However, one should note that we do not use any of the polarity label information of the dataset in our experiments but only for evaluating the performance of the JST model, as our model is fully unsupervised.

4.1 Preprocessing

Preprocessing was performed on the movie review data before the subsequent experiments. Firstly, punctuation, numbers and other non-alphabet characters were removed. Secondly, for the purpose of reducing the vocabulary size and addressing the issue of data sparseness, stemming was performed using the Porter's stemmer algorithm [17]. Stop words were also removed based on a stop word list[2]. After preprocessing, the corpus contains 2000 documents and 627,317 words with 25,166 distinct terms.

4.2 Defining Model Priors

As has been pointed out by Pang *et al.* [16], the sentiment classification problem is somehow more challenging than the traditional topic-based classification, since sentiment can be expressed in a more subtle manner while topics can be identified more easily according to the co-occurrence of keywords. One of the directions for improving the sentiment detection accuracy is to incorporate prior information or subjectivity lexicon (i.e., words bearing positive or negative polarity), which can be obtained in many different ways. Some approach annotates polarity to words based on manually constructed Appraisal Groups [24]. Other approach generates subjectivity lexicons in a semi-automatic manner [1]. More recently, Kaji and Kitsuregawa [9] proposed a method which can build polarity-tagged corpus from HTML documents fully automatically. While subjectivity lexicon generation is beyond the scope of this paper, here in our experiments, we investigated incorporating prior information obtained in four different ways into the JST and the tying-JST model, and explored how the prior information can improve the sentiment classification accuracy.

Paradigm word list The paradigm word list consists of a set of positive and negative words, e.g. *excellent* and *rubbish.* These lexicon words can be simply treated as the paradigms for defining the positive and negative semantic orientation, rather than for the purpose of training the algorithm [22].

The majority of the words were derived from the word lists used by Pang *et al.* [16] for their baseline result tests, with punctuation like '?' and '!' removed. However, we did notice the difference that the movie review data used by Pang *et al.* [16] is an older version with only 700 positive and 700 negative movie reviews, compared to the newer version we used that contains 1000 positive and 1000 negative documents. Hence, we added some additional paradigm words to the original list by reexamining a small portion of the corpus based on a very preliminary check of word frequency counts. Finally, the resulting paradigm word list contains 21 positive and 21 negative paradigm words respectively, as shown in Table 1.

Table 1: Paradigm word list.

Positive	dazzling brilliant phenomenal excellent fantastic gripping mesmerizing riveting spectacular cool awesome thrilling moving exciting love wonderful best great superb still beautiful
Negative	sucks terrible awful unwatchable hideous bad cliched boring stupid slow worst waste unexcit rubbish tedious unbearable pointless cheesy frustrated awkward disappointing

Mutual information (MI) In statistical language modeling, mutual information is a criterion widely used for calculating the semantic association between words. Here we use mutual information to select the words that have strong association with positive or negative sentiment classes. The top 20 words within each individual sentiment class were selected based on their MI scores and incorporated as prior information for our models.

Full subjectivity lexicon We also explored using the publicly available subjectivity word list with established polarities such as the MPQA subjectivity lexicon[3], which consists of 2718 positive and 4911 negative words[4]. By matching the words in the MPQA subjectivity lexicon with the vocabulary (with 25,166 distinct terms) of the movie review dataset, we finally obtained a subset of 1335 positive, 2214 negative words.

Filtered subjectivity lexicon The filtered subjectivity lexicon was obtained by removing from the full subjectivity lexicon the words occurred less than 50 times in the movie review dataset. The words whose polarity changed after stemming were also removed automatically. Finally, the filtered subjectivity lexicon contains 374 positive and 675 negative words.

Although one may argue that the paradigm word list and the MI extracted words seem requiring certain supervision information from the corpus itself, the subjectivity lexicon used here is fully domain-independent and does not bear any supervision information specifically to the movie review dataset. In fact, the JST model with the filtered subjectivity lexicon achieved better performance than the ones using the prior information obtained from paradigm word list or MI extracted words as can be seen later in Section 5. While it is well-known that sentiment classifiers trained on one domain often fail to produce satisfactory results in another domain, we speculate that the unsupervised nature of our JST model makes it highly portable to other domains.

4.3 Incorporating Prior Information

We modified Phan's GibbsLDA++ package[5] for the JST and tying-JST model implementation. In the experiments, the prior information was only utilized during the initialization of posterior distribution z, i.e. assignment of word token to sentiment label and topic. We chose a total number of 3 sentiment labels representing positive, negative and neutral, considering the fact that the sentiment of any word can be categorized into one of these three classes. The initialization starts by comparing each word token in the corpus against the words in the sentiment word list as described in Section 4.2. If there is a match, the word token is assigned with the corresponding sentiment label. Otherwise, a sentiment label is randomly sampled for a word token.

5. EXPERIMENTAL RESULTS

In this section, we will present and discuss the experimental results of both document sentiment classification and topic extraction, based on the movie review dataset.

5.1 Sentiment Classification

The document sentiment is classified based on $P(\mathbf{l}|d)$, the probability of sentiment label given document, which is approximated using Equation 8 in the implementation. In our

[2] http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words/

[3] http://www.cs.pitt.edu/mpqa/

[4] We discarded words with 'neutral' label in the subjectivity lexicon since the number of neutral words is small and many of the neutral words have multiple polarities, e.g. both neutral and positive.

[5] http://gibbslda.sourceforge.net/

Table 2: Results of incorporating various prior information.

Prior information	# of polarity words (pos./neg.)	JST (%)			Tying-JST (%)		
		pos.	neg.	overall	pos.	neg.	overall
Without prior information	0/0	63	56.6	59.8	59.2	53.8	56.5
Paradigm words	21/21	70.8	77.5	74.2	74.2	71.3	73.1
Paradigm words + MI	41/41	76.6	82.3	79.5	78	73.1	75.6
Full subjectivity lexicon	1335/2214	74.1	66.7	70.4	77.6	69	73.3
Filtered subjectivity lexicon	374/675	84.2	81.5	82.8	84.6	73.1	78.9
Filtered subjectivity lexicon (subjective MR)	374/675	96.2	73	84.6	89.2	74.8	82
Pang *et al.* (2002) [16]	N/A	Classifier used: SVMs			Best accuracy: 82.9%		
Pang and Lee (2004) [15] (subjective MR)	N/A	Classifier used: SVMs			Best accuracy: 87.2%		
Whitelaw *et al.* (2005) [24]	1597 appraisal groups	Classifier used: SVMs			Best accuracy: 90.2%		
Kennedy and Inkpen (2006) [10]	1955/2398	Classifier used: SVMs			Best accuracy: 86.2%		

experiments, we only consider the probability of positive and negative label given document, with the neutral label probability being ignored. There are two main reasons. Firstly, movie review sentiment classification in our case is effectively a binary classification problem, i.e. documents are being classified either as positive or negative, without the alternative of neutral. Secondly, the prior information we incorporated merely contributes to the positive and negative words, and consequently there will be much more influence on the probability distribution of positive and negative label given document, rather than the distribution of neutral label given document. Therefore, we define that a document d is classified as a positive-sentiment document if its probability of positive sentiment label given document $P(\mathbf{l_{pos}}|d)$, is greater than its probability of negative sentiment label given document $P(\mathbf{l_{neg}}|d)$, and vice versa.

In this section, we show how prior information improves the sentiment classification accuracy of the JST and tying-JST models and how topic mixtures affect the performance of our models.

5.1.1 Results with Different Prior Information

Table 2 shows the sentiment classification accuracy at document level by incorporating various prior information. The number of polarity (positive and negative) words in various subjectivity word list is also listed. In all of the results showed in the table, α is set to $\frac{50}{\#topics}$, β is set to 0.01. It should be noted that while LDA can produce reasonable results with a simple uniform Dirichlet prior for its hyperparameters, asymmetric prior γ for sentiment-document distribution should be used since it captures different correlations among sentiment labels. In our experiments, γ is set to 0.01 for positive sentiment label and 5 for negative sentiment label. The setting for γ was determined empirically. It is worth pointing out that hyperparameters can be learned from data directly by maximum likelihood or maximum a posteriori estimation [23]. Alternatively, an approximation approach such as moment matching could also be used to avoid iterative methods for the sake of simplicity and speed [12]. We leave the estimation of γ in a more principled way as future work.

It can be observed from Table 2 that without incorporating any prior information, JST only achieved around 60% overall accuracy. By incorporating merely 21 positive and 21 negative paradigm words, a significant performance improvement is observed with JST and tying-JST giving an overall of 74.2% and 73.1% accuracy respectively. We also experimented the combination of paradigm words and mutual information and evaluated how mutual information can help to improve the sentiment classification accuracy. We extracted the top 20 positive/negative words based on the MI value calculated from the 40 randomly selected labeled documents from the movie review dataset with equal number of positive and negative documents. Plus the paradigm words listed in Table 1, the total number of positive and negative words is 41 each. It can be observed that there is a considerable improvement in classification accuracy after incorporating the MI-extracted words, with 5.3% and 2.5% improvement for JST and tying-JST respectively.

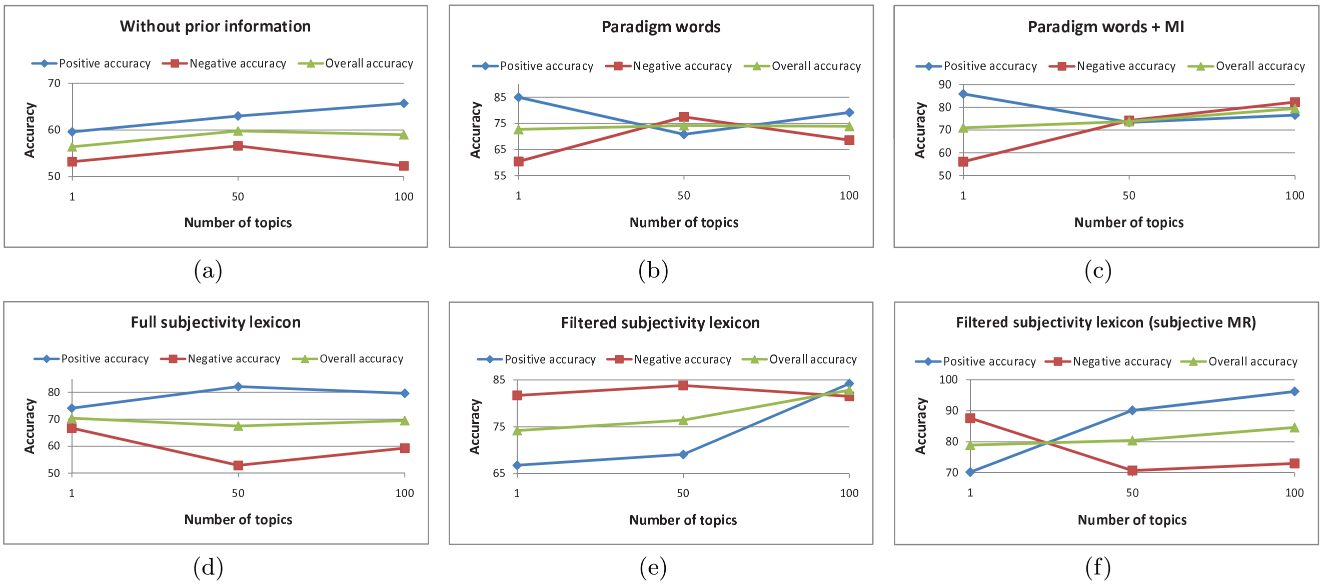
Subjectivity lexicons have attracted increasing focus in previous work [1]. Intuitively, one might expect that with a larger subjectivity lexicon and hence an increasing number of polarity words, sentiment classification performance would be improved since an overall polarity of a text can be inferred from the aggregated polarity of its individual words. However, the results shown in Table 2 reveal that incorporating the full subjectivity lexicon with 1335 positive and 2214 negative words in fact hurts the performance of both JST and tying-JST, with a relatively poor overall accuracy of 70.4% and 73.3% being achieved respectively. In contrast, with the filtered subjectivity lexicon by removing the infrequent polarity words, the performance of both models improves. Thus, the full subjectivity lexicon actually introduces more noise into the models and hence resulted in poorer performance. Also, the yielding results (82.8% for JST and 78.9% for tying-JST) are actually better than the performance by incorporating any aforementioned prior information.

We also observe that tying-JST performed consistently worse than the JST model except for the case of incorporating full subjectivity lexicon as prior information. Therefore, JST seems to be a more reasonable model design in terms of sentiment classification.

5.1.2 Results with Subjectivity Detection

In another set of experiments, we followed the approach in [15] and performed subjectivity detection (with sentences that do not express any opinions removed) prior to sentiment classification. Subjective sentences were extracted from the original movie review dataset using the LingPipe package[6]. First, we trained the subjectivity classifier based on the Sub-

[6] http://alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html



(a) (b) (c)

(d) (e) (f)

Figure 3: Sentiment classification accuracy VS. different topic numbers.

jectivity v1.0 dataset[7] which contains 5000 subjective and 5000 objective sentences. The trained classifier was then used to extract the subjective sentences from the movie review dataset, which reduces each single document to 5 to 25 sentences. After subjectivity detection and data preprocessing as described in Section 4.1, the dataset, which we named as "subjective MR", still contains 2000 documents but with a total of 334,336 words and 18,013 distinct terms (c.f. 25,166 distinct terms without subjectivity detection).

It can be seen from Table 2 that the best performance for both JST and tying-JST is obtained on the subjective MR dataset with the prior sentiment label information obtained from the filtered subjectivity lexicon, where an overall accuracy of 84.6% and 82% was achieved by JST and tying-JST respectively. This is a clear improvement over 82.8% and 78.9% when no subjectivity detection was performed. It suggests that though the subjective MR dataset is in a much compressed form, it is more effective than the full dataset as it retains comparable polarity information in a much cleaner way [15].

5.1.3 *Comparison with Existing Approaches*

For comparison, document-level sentiment classification results on the movie review dataset from four previous studies are also listed in the last four rows of Table 2. The best result reported in [16] is 82.9%, which is attained by support vector machines (SVMs) using bag-of-unigram features. The performance was later further improved to 87.2% [15] by applying SVMs on the subjective portions of the movie reviews which were extracted using a subjectivity detector as described in Section 5.1.2. Whitelaw *et al.* [24] used SVMs to train on the combination of different types of appraisal group features and the bag-of-words features for sentiment analysis. The reported best accuracy is 90.2% using 1,597 appraisal groups with each possible combination of Attitude and Orientation plus 48,314 bag-of-words features. Their appraisal groups were constructed semi-automatically and comprise of a total of 41,082 appraising groups. This is much more complicated than the subjectivity lexicon used in this paper. Kennedy and Inkpen [10] combined two main sources, General Inquirer[8] and *Choose the Right Word* [7], to obtain a total of 1,955 positive and 2,398 negative terms. They then trained two classifiers, one was based on counting the number of positive and negative terms contained in movie reviews and augmented with contextual valence shifters, while the other was based on SVMs trained from the combination of unigrams and valence shifter bigrams. These two classifiers were finally combined to give the best classification accuracy which is 86.2%.

In our experiment, the best overall accuracy achieved by JST is 84.6%, based on the filtered subjectivity lexicon and the subjective MR dataset. It outperforms the best result reported in [16] and is only 2.6% and 1.6% lower than the results reported in [15] and [10]. Even for the state-of-the-art result reported in [24], the best accuracy achieved by JST is only 5.6% lower. While all the previous studies mentioned here relied on the labeled movie review data to train sentiment classifiers, our proposed JST model is fully unsupervised. In addition, the previous reported results [15, 24, 10] were all based on 10-fold cross validation in a test set comprising of 200 documents only[9], our experimental results reported here are based on the whole movie review dataset with a total of 2000 documents.

5.1.4 *Results with Different Topics*

We also evaluated the mixture of topics and sentiments. Figure 3 shows the sentiment classification accuracy of the JST model incorporating prior information obtained in different ways with the number of topics set to 1, 50 and 100. When the topic number is set to 1, the JST model is es-

[7] http://www.cs.cornell.edu/People/pabo/movie-review-data/

[8] http://www.wjh.harvard.edu/∼inquirer/

[9] [16] used an early version of the movie review data which consists of 700 positive and 700 negative documents and the results were based on 3-fold cross validation.

Table 3: Example of topics extracted by JST under different sentiment labels.

Positive sentiment label						Negative sentiment label					
Topic 1		Topic 2		Topic 3		Topic 1		Topic 2		Topic 3	
w	$P(w\|z,l)$	w	$P(w\|z,l)$	w	$P(w\|z,l)$	w	$P(w\|z,l)$	w	$P(w\|z,l)$	w	$P(w\|z,l)$
good	0.084708	tom	0.035175	ship	0.059020	bad	0.079132	sex	0.065904	prison	0.073208
realli	0.046559	ryan	0.030281	titan	0.031586	worst	0.035402	scene	0.053660	evil	0.032196
plai	0.044174	hank	0.025388	crew	0.024439	plot	0.033687	sexual	0.031693	guard	0.031755
great	0.036645	comedi	0.021718	cameron	0.024439	stupid	0.029767	women	0.026291	green	0.029109
just	0.028990	star	0.020800	alien	0.022826	act	0.025602	rate	0.023770	hank	0.028227
perform	0.028362	drama	0.016519	jack	0.020751	suppos	0.025480	act	0.023230	wonder	0.027345
nice	0.026354	meg	0.015601	water	0.019137	script	0.024500	offens	0.018728	excute	0.026904
fun	0.025978	joe	0.014378	stori	0.017984	wast	0.024500	credict	0.016027	secret	0.025581
lot	0.025853	relationship	0.014072	rise	0.016601	dialogu	0.023643	porn	0.014587	mile	0.022936
act	0.022715	mail	0.013766	rose	0.013835	bore	0.022908	rape	0.013867	death	0.022495
direct	0.021586	blond	0.013460	boat	0.013374	poor	0.022908	femal	0.013686	base	0.022054
best	0.020331	run	0.012543	deep	0.013143	complet	0.020825	cut	0.013686	tom	0.019849
get	0.020331	phone	0.012237	ocean	0.012451	line	0.019968	gril	0.013506	convict	0.018967
entertain	0.018198	date	0.011931	board	0.011990	terribl	0.018988	parti	0.012426	return	0.018526
better	0.017445	got	0.011625	sink	0.011299	mess	0.015313	male	0.011886	franklin	0.016762
job	0.016692	busi	0.011319	sea	0.010838	wors	0.014333	bad	0.011346	happen	0.016321
talent	0.016064	cute	0.011013	rain	0.010838	dull	0.013598	nuditi	0.011166	power	0.014116
pretti	0.016064	sister	0.010708	dicaprio	0.010607	actor	0.012986	woman	0.010986	known	0.012352
try	0.015688	children	0.010096	storm	0.010377	total	0.012986	peopl	0.010986	instinct	0.011470
want	0.015186	dog	0.009790	disast	0.010146	isn	0.012863	nake	0.010625	inmat	0.011470

sentially transformed to a simple LDA model with only S topics, each of which corresponds to a sentiment label. Consequently, it ignores the correlation between sentiment labels and topics. It can be observed from Figure 3 that, JST performs worse with single topic compared to 50 and 100 topics, except for the case of full subjectivity lexicon as shown in Figure 3(d) where the single topic performance is almost the same as the one with 100 topics. For paradigm words + MI, filtered subjectivity lexicon and filter subjectivity lexicon (subjective MR) (Figures 3(c), 3(e), and 3(f)), the result with 100 topics outperforms the ones with other topic number settings. For the case when no prior information is applied as well as paradigm words as shown in Figure 3(a) and Figure 3(b), the results with 50 topics are almost the same as the ones achieved with 100 topics and both are higher than that of the single topic setting. It can be also easily seen that the results with filtered subjectivity lexicon in Figure 3(e) give the most balanced classification accuracy on both positive and negative documents. From the above, we can conclude that topic information indeed helps in sentiment classification as the JST model with the mixture of topics consistently outperforms a simple LDA model ignoring the mixture of topics. This justifies the proposal of our JST model. Also, the empirical results reveal that the optimum number of topics for the movie review dataset is 100.

5.2 Topic Extraction

The second goal of JST is to extract topics from the movie review dataset (without subjectivity detection) and evaluate the effectiveness of topic sentiment captured by the model. In the experiment, the distribution of words given topic and sentiment label was estimated using Equation (6). Unlike the LDA model that a word is drawn from the topic-word distribution, in JST one draws a word from the distribution over words conditioned on both topics and sentiment labels. Therefore, we analyze the extracted topics under two different sentiment labels (positive and negative). Six example topics extracted from the movie review dataset under positive and negative sentiment labels are shown in Table 3.

The three topics on the left columns of Table 3 were generated under the positive sentiment label and the remaining topics were generated under the negative sentiment label, each of which is represented by the top 20 topic words. As can be seen from the table that the six extracted topics are quite informative and coherent, where each of them tried to capture the underlying theme of a movie or the relevant comments from a movie reviewer. For example, under the positive sentiment label category, topic 1 is likely to be very positive review comments for a movie; topic 2 is apparently about the movie *"You've got a mail"* by Tom Hanks and Meg Ryan; topic 3 is closely related to the very popular romantic movie *"Titanic"* directed by James Cameron and casted by Leonardo DiCaprio and Kate Winslet. For the topics under the negative sentiment category, topic 1 is probably the criticism made by a movie reviewer, while topic 2 is about movies related to sex/porn issues and topic 3 is likely to be the movie *"Green Mile"* by Tom Hanks.

In terms of topic sentiment, by examining each of the topics in Table 3, it is quite evident that topic 1 under the positive sentiment label and topic 1 under the negative label indeed bear positive and negative sentiment respectively. For topic 2 and topic 3 under the negative sentiment label, it is still fairly easy to recognize that some of their topic words convey negative sentiments though not as strong as the ones in topic 1. Topic 2 and topic 3 under the positive sentiment label mainly describe movie plots with less words carrying positive sentiment compared to topic 1 under the same category. Manually examining the data reveals that the terms that seem not conveying sentiments under these two topics in fact appear in the context expressing positive sentiments. The above analysis illustrates the effectiveness of JST in extracting mixture of topics from a corpus.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a joint sentiment/topic (JST) model which can detect document level sentiment and extract mixture of topics from text simultaneously. In contrast to most of the existing approaches in sentiment classification which rely on supervised learning, the proposed JST model is fully unsupervised, thus provides more flexibilities and can be easier adapted to other applications. Experiments have been conducted to evaluate the performance of JST based on the movie review dataset. The preliminary results demonstrated that our model is able to give competitive performance in document level sentiment classification compared with the results generated by other existing supervised approaches and the discovered topics are indeed coherent and informative.

One of the limitations of our model is that it represents each document as a bag of words and thus ignores the word ordering. It will probably predict the sentiment of "not good movie" being positive and the sentiment of "not bad movie" being negative. Thus, in future work, we will extend the model to include higher order information (bigrams or trigrams). Another promising future step is to extend JST to detect the polarity of text at various granularity levels, e.g. detecting sentiment labels for more fine-grained topics. We also intend to carry out a large scale of experiments and evaluate the model performance on datasets from different domains.

7. ACKNOWLEDGMENTS

We thank Rui Wang and Naihui He for processing part of the prior information and Lei Wang for providing computing resources.

8. REFERENCES

[1] A. Abbasi, H. Chen, and A. Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26(3):1–34, 2008.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[3] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[4] Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[5] K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 345–354, Sydney, Australia, July 2006. Association for Computational Linguistics.

[6] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(90001):5228–5235, 2004.

[7] S. Hayakawa and E. Ehrlich. *Choose the right word: A contemporary guide to selecting the precise word for every situation.* Collins, 1994.

[8] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA, 1999. ACM.

[9] N. Kaji and M. Kitsuregawa. Automatic construction of polarity-tagged corpus from html documents. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 452–459, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[10] A. Kennedy and D. Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125, 2006.

[11] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1367, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[12] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 577–584, 2006.

[13] R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 432–439, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[14] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 171–180, New York, NY, USA, 2007. ACM.

[15] B. Pang and L. Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 271, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[16] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[17] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[18] M. Steyvers and T. Griffiths. Probabilistic Topic Models. *Handbook of Latent Semantic Analysis*, page 427, 2007.

[19] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[20] I. Titov and R. McDonald. Modeling online reviews

with multi-grain topic models. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 111–120, New York, NY, USA, 2008. ACM.

[21] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[22] P. D. Turney and M. L. Littman. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. *CoRR*, cs.LG/0212012, 2002.

[23] H. M. Wallach. Topic modeling: beyond bag-of-words. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 977–984, 2006.

[24] C. Whitelaw, N. Garg, and S. Argamon. Using appraisal groups for sentiment analysis. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631, New York, NY, USA, 2005. ACM.

[25] J. Zhao, K. Liu, and G. Wang. Adding redundant features for CRFs-based sentence sentiment classification. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 117–126, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

Journal of Statistical Software

May 2011, Volume 40, Issue 13. http://www.jstatsoft.org/

topicmodels: An R Package for Fitting Topic Models

Bettina Grün
Johannes Kepler Universität Linz

Kurt Hornik
WU Wirtschaftsuniversität Wien

Abstract

Topic models allow the probabilistic modeling of term frequency occurrences in documents. The fitted model can be used to estimate the similarity between documents as well as between a set of specified keywords using an additional layer of latent variables which are referred to as topics. The R package **topicmodels** provides basic infrastructure for fitting topic models based on data structures from the text mining package **tm**. The package includes interfaces to two algorithms for fitting topic models: the variational expectation-maximization algorithm provided by David M. Blei and co-authors and an algorithm using Gibbs sampling by Xuan-Hieu Phan and co-authors.

Keywords: Gibbs sampling, R, text analysis, topic model, variational EM.

1. Introduction

In machine learning and natural language processing topic models are generative models which provide a probabilistic framework for the term frequency occurrences in documents in a given corpus. Using only the term frequencies assumes that the information in which order the words occur in a document is negligible. This assumption is also referred to as the *exchangeability* assumption for the words in a document and this assumption leads to bag-of-words models.

Topic models extend and build on classical methods in natural language processing such as the unigram model and the mixture of unigram models (Nigam, McCallum, Thrun, and Mitchell 2000) as well as Latent Semantic Analysis (LSA; Deerwester, Dumais, Furnas, Landauer, and Harshman 1990). Topic models differ from the unigram or the mixture of unigram models because they are mixed-membership models (see for example Airoldi, Blei, Fienberg, and Xing 2008). In the unigram model each word is assumed to be drawn from the same term distribution, in the mixture of unigram models a topic is drawn for each document and all words in a document are drawn from the term distribution of the topic. In mixed-membership models documents are not assumed to belong to single topics, but to simultaneously belong to several topics and the topic distributions vary over documents.

An early topic model was proposed by Hofmann (1999) who developed probabilistic LSA. He assumed that the interdependence between words in a document can be explained by the latent topics the document belongs to. Conditional on the topic assignments of the words the word occurrences in a document are independent. The latent Dirichlet allocation (LDA; Blei, Ng, and Jordan 2003b) model is a Bayesian mixture model for discrete data where topics are assumed to be uncorrelated. The correlated topics model (CTM; Blei and Lafferty 2007) is an extension of the LDA model where correlations between topics are allowed. An introduction to topic models is given in Steyvers and Griffiths (2007) and Blei and Lafferty (2009). Topic models have previously been used for a variety of applications, including ad-hoc information retrieval (Wei and Croft 2006), geographical information retrieval (Li, Wang, Xie, Wang, and Ma 2008) and the analysis of the development of ideas over time in the field of computational linguistics (Hall, Jurafsky, and Manning 2008).

C code for fitting the LDA model (`http://www.cs.princeton.edu/~blei/lda-c/`) and the CTM (`http://www.cs.princeton.edu/~blei/ctm-c/`) is available under the GPL from David M. Blei and co-authors, who introduced these models in their papers. The method used for fitting the models is the variational expectation-maximization (VEM) algorithm. Other implementations for fitting topic models—especially of the LDA model—are available. The standalone program lda (Mochihashi 2004a,b) provides standard VEM estimation. An implementation in Python of an online version of LDA using VEM estimation as described in Hoffman, Blei, and Bach (2010) is available under the GPL from the first author's web page (`http://www.cs.princeton.edu/~mdhoffma/`). For Bayesian estimation using Gibbs sampling several implementations are available. **GibbsLDA++** (Phan, Nguyen, and Horiguchi 2008) is available under the GPL from `http://gibbslda.sourceforge.net/`. The **MATLAB Topic Modeling** toolbox (Griffiths and Steyvers 2004; Steyvers and Griffiths 2011) is free for scientific use. A license must be obtained from the authors to use it for commercial purposes. **MALLET** (McCallum 2002) is released under the CPL and is a Java-based package which is more general in allowing for statistical natural language processing, document classification, clustering, topic modeling using LDA, information extraction, and other machine learning applications to text. A general toolkit for implementing hierarchical Bayesian models is provided by the Hierarchical Bayes compiler **HBC** (Daumé III 2008), which also allows to fit the LDA model. Another general framework for running Bayesian inference in graphical models which allows to fit the LDA model is available through **Infer.NET** (Microsoft Corporation 2010). The fast collapsed Gibbs sampling method is described in Porteous, Asuncion, Newman, Ihler, Smyth, and Welling (2008) and code is also available from the first author's web page (`http://www.ics.uci.edu/~iporteou/fastlda/`).

For R, an environment for statistical computing and graphics (R Development Core Team 2011), the Comprehensive R Archive Network (`http://CRAN.R-project.org/`) features two packages for fitting topic models: **topicmodels** and **lda**. The R package **lda** (Chang 2010) provides collapsed Gibbs sampling methods for LDA and related topic model variants, with the Gibbs sampler implemented in C. All models in package **lda** are fitted using Gibbs sampling for determining the posterior probability of the latent variables. Wrappers for the expectation-maximization (EM) algorithm are provided which build on this functionality for the E-step. Note that this implementation therefore differs in general from the estimation technique proposed in the original papers introducing these model variants, where the VEM algorithm is usually applied.

The R package **topicmodels** currently provides an interface to the code for fitting an LDA model and a CTM with the VEM algorithm as implemented by Blei and co-authors and to the code for fitting an LDA topic model with Gibbs sampling written by Phan and co-authors. Package **topicmodels** builds on package **tm** (Feinerer, Hornik, and Meyer 2008; Feinerer 2011) which constitutes a framework for text mining applications within R. **tm** provides infrastructure for constructing a corpus, e.g., by reading in text data from PDF files, and transforming a corpus to a document-term matrix which is the input data for topic models. In package **topicmodels** the respective code is directly called through an interface at the C level avoiding file input and output, and hence substantially improving performance. The functionality for data input and output in the original code was substituted and R objects are directly used as input and S4 objects as output to R. The same main function allows fitting the LDA model with different estimation methods returning objects only slightly different in structure. In addition the strategies for model selection and inference are applicable in both cases. This allows for easy use and comparison of both current state-of-the-art estimation techniques for topic models. Packages **topicmodels** aims at extensibility by providing an interface for inclusion of other estimation methods of topic models.

This paper is structured as follows: Section 2 introduces the specification of topic models, outlines the estimation with the VEM as well as Gibbs sampling and gives an overview of pre-processing steps and methods for model selection and inference. The main fitter functions in the package and the helper functions for analyzing a fitted model are presented in Section 3. An illustrative example for using the package is given in Section 4 where topic models are fitted to the corpus of abstracts in the *Journal of Statistical Software*. A further example is presented in Section 5 using a subset of the Associated Press data set, a larger subset of which was also analyzed in Blei *et al.* (2003b). This data set consists of documents which focus on different content areas and while still being rather small has similar characteristics as other corpora used in the topic models literature. Finally, extending the package to new estimation methods is described in Section 6 using package **rjags** (Plummer 2011).

2. Topic model specification and estimation

2.1. Model specification

For both models—LDA and CTM—the number of topics k has to be fixed a-priori. The LDA model and the CTM assume the following generative process for a document $w = (w_1, \ldots, w_N)$ of a corpus D containing N words from a vocabulary consisting of V different terms, $w_i \in \{1, \ldots, V\}$ for all $i = 1, \ldots, N$.

For LDA the generative model consists of the following three steps.

Step 1: The term distribution β is determined for each topic by

$$\beta \sim \text{Dirichlet}(\delta).$$

Step 2: The proportions θ of the topic distribution for the document w are determined by

$$\theta \sim \text{Dirichlet}(\alpha).$$

Step 3: For each of the N words w_i

(a) Choose a topic $z_i \sim \text{Multinomial}(\theta)$.

(b) Choose a word w_i from a multinomial probability distribution conditioned on the topic z_i: $p(w_i|z_i, \beta)$.

β is the term distribution of topics and contains the probability of a word occurring in a given topic.

For CTM Step 2 is modified to

Step 2a: The proportions θ of the topic distribution for the document w are determined by drawing

$$\eta \sim N(\mu, \Sigma)$$

with $\eta \in \mathbb{R}^{(k-1)}$ and $\Sigma \in \mathbb{R}^{(k-1)\times(k-1)}$.

Set $\tilde{\eta}^\top = (\eta^\top, 0)$. θ is given by

$$\theta_K = \frac{\exp\{\tilde{\eta}_K\}}{\sum_{i=1}^k \exp\{\tilde{\eta}_i\}}$$

for $K = 1, \ldots, k$.

2.2. Estimation

For maximum likelihood (ML) estimation of the LDA model the log-likelihood of the data, i.e., the sum over the log-likelihoods of all documents, is maximized with respect to the model parameters α and β. In this setting β and not δ is in general the parameter of interest. For the CTM model the log-likelihood of the data is maximized with respect to the model parameters μ, Σ and β. For VEM estimation the log-likelihood for one document $w \in D$ is for LDA given by

$$\begin{aligned}\ell(\alpha, \beta) &= \log\left(p(w|\alpha, \beta)\right)\\ &= \log \int \left\{\sum_z \left[\prod_{i=1}^N p(w_i|z_i, \beta)p(z_i|\theta)\right]\right\} p(\theta|\alpha)d\theta\end{aligned}$$

and for CTM by

$$\begin{aligned}\ell(\mu, \Sigma, \beta) &= \log\left(p(w|\mu, \Sigma, \beta)\right)\\ &= \log \int \left\{\sum_z \left[\prod_{i=1}^N p(w_i|z_i, \beta)p(z_i|\theta)\right]\right\} p(\theta|\mu, \Sigma)d\theta.\end{aligned}$$

The sum over $z = (z_i)_{i=1,\ldots,N}$ includes all combinations of assigning the N words in the document to the k topics.

The quantities $p(w|\alpha, \beta)$ for the LDA model and $p(w|\mu, \Sigma, \beta)$ for the CTM cannot be tractably computed. Hence, a VEM procedure is used for estimation. The EM algorithm (Dempster, Laird, and Rubin 1977) is an iterative method for determining an ML estimate in a missing data framework where the complete likelihood of the observed and missing data is easier to maximize than the likelihood of the observed data only. It iterates between an expectation (E)-step where the expected complete likelihood given the data and current parameter estimates is determined and a maximization (M)-step where the expected complete likelihood is maximized to find new parameter estimates. For topic models the missing data in the EM algorithm are the latent variables θ and z for LDA and η and z for CTM.

For topic models a VEM algorithm is used instead of an ordinary EM algorithm because the expected complete likelihood in the E-step is still computationally intractable. For an introduction into variational inference see for example Wainwright and Jordan (2008). To facilitate the E-step the posterior distribution $p(\theta, z|w, \alpha, \beta)$ is replaced by a variational distribution $q(\theta, z|\gamma, \phi)$. This implies that in the E-step instead of

$$\mathsf{E}_p[\log p(\theta, z|w, \alpha, \beta)]$$

the following is determined

$$\mathsf{E}_q[\log p(\theta, z|w, \alpha, \beta)].$$

The parameters for the variational distributions are document specific and hence are allowed to vary over documents which is not the case for α and β. For the LDA model the variational parameters γ and ϕ for a given document w are determined by

$$(\gamma^*, \phi^*) = \arg\min_{(\gamma,\phi)} \mathrm{D}_{\mathrm{KL}}(q(\theta, z|\gamma, \phi)||p(\theta, z|w, \alpha, \beta)).$$

D_{KL} denotes the Kullback-Leibler (KL) divergence. The variational distribution is set equal to

$$q(\theta, z|\gamma, \phi) = q_1(\theta|\gamma) \prod_{i=1}^{N} q_2(z_i|\phi_i),$$

where $q_1()$ is a Dirichlet distribution with parameters γ and $q_2()$ is a multinomial distribution with parameters ϕ_i. Analogously for the CTM the variational parameters are determined by

$$(\lambda^*, \nu^*, \phi^*) = \arg\min_{(\lambda,\nu,\phi)} \mathrm{D}_{\mathrm{KL}}(q(\eta, z|\lambda, \nu^2, \phi)||p(\eta, z|w, \mu, \Sigma, \beta)).$$

Since the variational parameters are fitted separately for each document the variational covariance matrix can be assumed to be diagonal. The variational distribution is set to

$$q(\eta, z|\lambda, \nu^2, \phi) = \prod_{K=1}^{k-1} q_1(\eta_K|\lambda_K, \nu_K^2) \prod_{i=1}^{N} q_2(z_i|\phi_i),$$

where $q_1()$ is a univariate Gaussian distribution with mean λ_K and variance ν_K^2, and $q_2()$ again denotes a multinomial distribution with parameters ϕ_i. Using this simple model for η has the advantage that it is computationally less demanding while still providing enough flexibility. Over all documents this leads to a mixture of normal distributions with diagonal variance-covariance matrices. This mixture distribution allows to approximate the marginal distribution over all documents which has an arbitrary variance-covariance matrix.

For the LDA model it can be shown with the following equality that the variational parameters result in a lower bound for the log-likelihood

$$\log p(w|\alpha,\beta) = L(\gamma,\phi;\alpha,\beta) + \mathrm{D_{KL}}(q(\theta,z|\gamma,\phi)||p(\theta,z|w,\alpha,\beta))$$

where

$$L(\gamma,\phi;\alpha,\beta) = \mathsf{E}_q[\log p(\theta,z,w|\alpha,\beta)] - \mathsf{E}_q[\log q(\theta,z)]$$

(see Blei *et al.* 2003b, p. 1019). Maximizing the lower bound $L(\gamma,\phi;\alpha,\beta)$ with respect to γ and ϕ is equivalent to minimizing the KL divergence between the variational posterior probability and the true posterior probability. This holds analogously for the CTM.

For estimation the following steps are repeated until convergence of the lower bound of the log-likelihood.

E-step: For each document find the optimal values of the variational parameters $\{\gamma,\phi\}$ for the LDA model and $\{\lambda,\nu,\phi\}$ for the CTM.

M-step: Maximize the resulting lower bound on the log-likelihood with respect to the model parameters α and β for the LDA model and μ, Σ and β for the CTM.

For inference the latent variables θ and z are often of interest to determine which topics a document consists of and which topic a certain word in a document was drawn from. Under the assumption that the variational posterior probability is a good approximation of the true posterior probability it can be used to determine estimates for the latent variables. In the following inference is always based on the variational posterior probabilities if the VEM is used for estimation.

For Gibbs sampling in the LDA model draws from the posterior distribution $p(z|w)$ are obtained by sampling from

$$p(z_i = K|w, z_{-i}) \propto \frac{n^{(j)}_{-i,K} + \delta}{n^{(\cdot)}_{-i,K} + V\delta} \frac{n^{(d_i)}_{-i,K} + \alpha}{n^{(d_i)}_{-i,\cdot} + k\alpha}$$

(see Griffiths and Steyvers 2004; Phan *et al.* 2008). z_{-i} is the vector of current topic memberships of all words without the ith word w_i. The index j indicates that w_i is equal to the jth term in the vocabulary. $n^{(j)}_{-i,K}$ gives how often the jth term of the vocabulary is currently assigned to topic K without the ith word. The dot . implies that summation over this index is performed. d_i indicates the document in the corpus to which word w_i belongs. In the Bayesian model formulation δ and α are the parameters of the prior distributions for the term distribution of the topics β and the topic distribution of documents θ, respectively. The predictive distributions of the parameters θ and β given w and z are given by

$$\hat{\beta}^{(j)}_K = \frac{n^{(j)}_K + \delta}{n^{(\cdot)}_K + V\delta}, \qquad \hat{\theta}^{(d)}_K = \frac{n^{(d)}_K + \alpha}{n^{(\cdot)}_K + k\alpha},$$

for $j = 1,\ldots,V$ and $d = 1,\ldots,D$.

2.3. Pre-processing

The input data for topic models is a document-term matrix. The rows in this matrix correspond to the documents and the columns to the terms. The entry m_{ij} indicates how often the jth term occurred in the ith document. The number of rows is equal to the size of the corpus and the number of columns to the size of the vocabulary. The data pre-processing step involves selecting a suitable vocabulary, which corresponds to the columns of the document-term matrix. Typically, the vocabulary will not be given a-priori, but determined using the available data. The mapping from the document to the term frequency vector involves tokenizing the document and then processing the tokens for example by converting them to lower-case, removing punctuation characters, removing numbers, stemming, removing stop words and omitting terms with a length below a certain minimum. In addition the final document-term matrix can be reduced by selecting only the terms which occur in a minimum number of documents (see Griffiths and Steyvers 2004, who use a value of 5) or those terms with the highest term-frequency inverse document frequency (tf-idf) scores (Blei and Lafferty 2009). The tf-idf scores are only used for selecting the vocabulary, the input data consisting of the document-term matrix uses a term-frequency weighting.

2.4. Model selection

For fitting the LDA model or the CTM to a given document-term matrix the number of topics needs to be fixed a-priori. Additionally, estimation using Gibbs sampling requires specification of values for the parameters of the prior distributions. Griffiths and Steyvers (2004) suggest a value of $50/k$ for α and 0.1 for δ. Because the number of topics is in general not known, models with several different numbers of topics are fitted and the optimal number is determined in a data-driven way. Model selection with respect to the number of topics is possible by splitting the data into training and test data sets. The likelihood for the test data is then approximated using the lower bound for VEM estimation. For Gibbs sampling the log-likelihood is given by

$$\log(p(w|z)) = k \log\left(\frac{\Gamma(V\delta)}{\Gamma(\delta)^V}\right) + \sum_{K=1}^{k} \left\{ \left[\sum_{j=1}^{V} \log(\Gamma(n_K^{(j)} + \delta))\right] - \log(\Gamma(n_K^{(.)} + V\delta)) \right\}.$$

The perplexity is often used to evaluate the models on held-out data and is equivalent to the geometric mean per-word likelihood.

$$\text{Perplexity}(w) = \exp\left\{ -\frac{\log(p(w))}{\sum_{d=1}^{D}\sum_{j=1}^{V} n^{(jd)}} \right\}$$

$n^{(jd)}$ denotes how often the jth term occurred in the dth document. If the model is fitted using Gibbs sampling the likelihood is determined for the perplexity using

$$\log(p(w)) = \sum_{d=1}^{D}\sum_{j=1}^{V} n^{(jd)} \log\left[\sum_{K=1}^{k} \theta_K^{(d)} \beta_K^{(j)}\right]$$

(see Newman, Asuncion, Smyth, and Welling 2009). The topic weights $\theta_K^{(d)}$ can either be determined for the new data using Gibbs sampling where the term distributions for topics are kept fixed or equal weights are used as implied by the prior distribution. If the perplexity is calculated by averaging over several draws the mean is taken over the samples inside the logarithm.

In addition the marginal likelihoods of the models with different numbers of topics can be compared for model selection if Gibbs sampling is used for model estimation. Griffiths and Steyvers (2004) determine the marginal likelihood using the harmonic mean estimator (Newton and Raftery 1994), which is attractive from a computational point of view because it only requires the evaluation of the log-likelihood for the different posterior draws of the parameters. The drawback however is that the estimator might have infinite variance.

Different methods for evaluating fitted topic models on held-out documents are discussed and compared in Wallach, Murray, Salakhutdinov, and Mimno (2009). Another possibility for model selection is to use hierarchical Dirichlet processes as suggested in Teh, Jordan, Beal, and Blei (2006).

3. Application: Main functions `LDA()` and `CTM()`

The main functions in package **topicmodels** for fitting the LDA and CTM models are `LDA()` and `CTM()`, respectively.

```
LDA(x, k, method = "VEM", control = NULL, model = NULL, ...)
CTM(x, k, method = "VEM", control = NULL, model = NULL, ...)
```

These two functions have the same arguments. `x` is a suitable document-term matrix with non-negative integer count entries, typically a `"DocumentTermMatrix"` as obtained from package **tm**. Internally, **topicmodels** uses the simple triplet matrix representation of package **slam** (Hornik, Meyer, and Buchta 2011)—which, similar to the "coordinate list" (COO) sparse matrix format, stores the information about non-zero entries x_{ij} in the form of (i, j, x_{ij}) triplets. `x` can be any object coercible to such simple triplet matrices (with count entries), in particular objects obtained from readers for commonly employed document-term matrix storage formats. For example the reader `read_dtm_Blei_et_al()` available in package **tm** allows to read in data provided in the format used for the code by Blei and co-authors. `k` is an integer (larger than 1) specifying the number of topics. `method` determines the estimation method used and currently can be either `"VEM"` or `"Gibbs"` for `LDA()` and only `"VEM"` for `CTM()`. Users can provide their own fit functions to use a different estimation technique or fit a slightly different model variant and specify them to be called within `LDA()` and `CTM()` via the `method` argument. Argument `model` allows to provide an already fitted topic model which is used to initialize the estimation.

Argument `control` can be either specified as a named list or as a suitable S4 object where the class depends on the chosen method. In general a user will provide named lists and coercion to an S4 object will internally be performed. The following arguments are possible for the control for fitting the LDA model with the VEM algorithm. They are set to their default values.

```
R> control_LDA_VEM <- list(
+    estimate.alpha = TRUE, alpha = 50/k, estimate.beta = TRUE,
+    verbose = 0, prefix = tempfile(), save = 0, keep = 0,
+     seed = as.integer(Sys.time()), nstart = 1, best = TRUE,
+     var = list(iter.max = 500, tol = 10^-6),
+     em = list(iter.max = 1000, tol = 10^-4),
+     initialize = "random")
```

The arguments are described in detail below.

`estimate.alpha`, `alpha`, `estimate.beta`: By default α is estimated (`estimate.alpha = TRUE`) and the starting value for α is $50/k$ as suggested by Griffiths and Steyvers (2004). If α is not estimated, it is held fixed at the initial value. If the term distributions for the topics are already given by a previously fitted model, only the topic distributions for documents can be estimated using `estimate.beta = FALSE`. This is useful for example if a fitted model is evaluated on hold-out data or for new data.

`verbose`, `prefix`, `save`, `keep`: By default no information is printed during the algorithm (`verbose = 0`). If `verbose` is a positive integer every `verbose` iteration information is printed. `save` equal to 0 indicates that no intermediate results are saved in files with prefix `prefix`. If equal to a positive integer, every `save` iterations intermediate results are saved. If `keep` is a positive integer, the log-likelihood values are stored every `keep` iteration.

`seed`, `nstart`, `best`: For reproducibility a random seed can be set which is used in the external code. `nstart` indicates the number of repeated runs with random initializations. `seed` needs to have the length `nstart`. If `best = TRUE` only the best model over all runs with respect to the log-likelihood is returned.

`var`, `em`: These arguments control how convergence is assessed for the variational inference step and for the EM algorithm steps by setting a maximum number of iterations (`iter.max`) and a tolerance for the relative change in the likelihood (`tol`). If during the EM algorithm the likelihood is not increased in one step, the maximum number of iterations in the variational inference step is doubled.

If the maximum number of iterations is set to `-1` in the variational inference step, there is no bound on the number of iterations and the algorithm continues until the tolerance criterion is met. If the maximum number of iterations is `-1` for the EM algorithm, no M-step is made and only the variational inference is optimized. This is useful if the variational parameters should be determined for new documents. The default values for the convergence checks are chosen similar to those suggested in the code available from Blei's web page as additional material to Blei *et al.* (2003b) and Blei and Lafferty (2007).

`initialize`: This parameter determines how the topics are initialized and can be either equal to `"random"`, `"seeded"` or `"model"`. Random initialization means that each topic is initialized randomly, seeded initialization signifies that each topic is initialized to a distribution smoothed from a randomly chosen document. If `initialize = "model"` a fitted model needs to be provided which is used for initialization, otherwise random initialization is used.

The possible arguments controlling how the LDA model is fitted using Gibbs sampling are given below together with their default values.

```
R> control_LDA_Gibbs <- list(
+    alpha = 50/k, estimate.beta = TRUE, verbose = 0, prefix = tempfile(),
+    save = 0, keep = 0, seed = as.integer(Sys.time()), nstart = 1,
+    best = TRUE, delta = 0.1, iter = 2000, burnin = 0, thin = 2000)
```

`alpha`, `estimate.beta`, `verbose`, `prefix`, `save`, `keep`, `seed` and `nstart` are the same as for estimation with the VEM algorithm. The other parameters are described below in detail.

`delta`: This parameter specifies the parameter of the prior distribution of the term distribution over topics. The default 0.1 is suggested in Griffiths and Steyvers (2004).

`iter`, `burnin`, `thin`: These parameters control how many Gibbs sampling draws are made. The first `burnin` iterations are discarded and then every `thin` iteration is returned for `iter` iterations.

`best`: All draws are returned if `best = FALSE`, otherwise only the draw with the highest posterior likelihood over all runs is returned.

For the CTM model using the VEM algorithm the following arguments can be used to control the estimation.

```
R> control_CTM_VEM <- list(
+    estimate.beta = TRUE, verbose = 0, prefix = tempfile(), save = 0,
+    keep = 0, seed = as.integer(Sys.time()), nstart = 1L, best = TRUE,
+    var = list(iter.max = 500, tol = 10^-6),
+    em = list(iter.max = 1000, tol = 10^-4), initialize = "random",
+    cg = list(iter.max = 500, tol = 10^-5))
```

`estimate.beta`, `verbose`, `prefix`, `save`, `keep`, `seed`, `nstart`, `best`, `var`, `em` and `initialize` are the same as for VEM estimation of the LDA model. If the log-likelihood is decreased in an E-step, the maximum number of iterations in the variational inference step is increased by 10 or, if no maximum number is set, the tolerance for convergence is divided by 10 and the same E-step is continued. The only additional argument is `cg`.

`cg`: This controls how many iterations at most are used (`iter.max`) and how convergence is assessed (`tol`) in the conjugate gradient step in fitting the variational mean and variance per document.

`LDA()` and `CTM()` return S4 objects of a class which inherits from `"TopicModel"` (or a list of objects inheriting from class `"TopicModel"` if `best = FALSE`). Because of certain differences in the fitted objects there are sub-classes with respect to the model fitted (LDA or CTM) and the estimation method used (VEM or Gibbs sampling). The class `"TopicModel"` contains the call, the dimension of the document-term matrix, the number of words in the document-term matrix, the control object, the number of topics and the terms and document names and the number of iterations made. The estimates for the topic distributions for the documents are included which are the estimates of the corresponding variational parameters for the VEM algorithm and the parameters of the predictive distributions for Gibbs sampling. The term distribution of the topics are also contained which are the ML estimates for the VEM algorithm and the parameters of the predictive distributions for Gibbs sampling. In additional slots the objects contain the assignment of terms to the most likely topic and the log-likelihood which is $\log p(w|\alpha, \beta)$ for LDA with VEM estimation, $\log p(w|z)$ for LDA using Gibbs sampling and $\log p(w|\mu, \Sigma, \beta)$ for CTM with VEM estimation. For VEM estimation the log-likelihood is returned separately for each document. If a positive `keep` control argument was given, the log-likelihood values of every `keep` iteration is contained. The extending class `"LDA"` has an additional slot for α, `"CTM"` additional slots for μ and Σ. `"LDA_Gibbs"` which extends class `"LDA"` has a slot for δ and `"CTM_VEM"` which extends `"CTM"` has an additional slot for ν^2.

Helper functions to analyze the fitted models are available. `logLik()` obtains the log-likelihood of the fitted model and `perplexity()` can be used to determine the perplexity of a fitted model also for new data. `posterior()` allows one to obtain the topic distributions for documents and the term distributions for topics. There is a `newdata` argument which needs to be given a document-term matrix and where the topic distributions for these new documents are determined without fitting the term distributions of topics. Finally, functions `terms()` and `topics()` allow to obtain from a fitted topic model either the `k` most likely terms for topics or topics for documents respectively, or all terms for topics or topics for documents where the probability is above the specified `threshold`.

4. Illustrative example: Abstracts of JSS papers

The application of the package **topicmodels** is demonstrated on the collection of abstracts of the *Journal of Statistical Software* (JSS) up to 2010-08-05. This illustrative application has the advantage that the analysis can be reproduced in an interactive way and each of the commands can easily be tried out. By contrast re-estimating the models of the application in Section 5 using the Associated Press data would take too long for such a purpose. But Section 5 provides a rather genuine medium-to-large scale application.

The JSS data is available as a list matrix in the package **corpus.JSS.papers** which can be installed and loaded by

```
R> install.packages("corpus.JSS.papers",
+    repos = "http://datacube.wu.ac.at/", type = "source")
R> data("JSS_papers", package = "corpus.JSS.papers")
```

Alternatively, one can harvest JSS publication Dublin Core (http://dublincore.org/) metadata (including information on authors, publication date and the abstract) from the JSS web site using the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), for which package **OAIHarvester** (Hornik 2011) provides an R client.

```
R> library("OAIHarvester")
R> x <- oaih_list_records("http://www.jstatsoft.org/oai")
R> JSS_papers <- oaih_transform(x[, "metadata"])
R> JSS_papers <- JSS_papers[order(as.Date(unlist(JSS_papers[, "date"]))), ]
R> JSS_papers <- JSS_papers[grep("Abstract:", JSS_papers[, "description"]), ]
R> JSS_papers[, "description"] <- sub(".*\nAbstract:\n", "",
+    unlist(JSS_papers[, "description"]))
```

For reproducibility of results we use only abstracts published up to 2010-08-05 and omit those containing non-ASCII characters in the abstracts.

```
R> JSS_papers <- JSS_papers[JSS_papers[,"date"] < "2010-08-05",]
R> JSS_papers <- JSS_papers[sapply(JSS_papers[, "description"],
+    Encoding) == "unknown",]
```

The final data set contains 348 documents. Before analysis we transform it to a `"Corpus"` using package **tm**. HTML markup in the abstracts for greek letters, subscripting, etc., is removed using package **XML** (Temple Lang 2010).

```
R> library("topicmodels")
R> library("XML")
R> remove_HTML_markup <- function(s) {
+    doc <- htmlTreeParse(s, asText = TRUE, trim = FALSE)
+    xmlValue(xmlRoot(doc))
+  }
R> corpus <- Corpus(VectorSource(sapply(JSS_papers[, "description"],
+    remove_HTML_markup)))
```

The corpus is exported to a document-term matrix using function `DocumentTermMatrix()` from package **tm**. The terms are stemmed and the stop words, punctuation, numbers and terms of length less than 3 are removed using the `control` argument. (We use a `C` locale for reproducibility.)

```
R> Sys.setlocale("LC_COLLATE", "C")

[1] "C"

R> JSS_dtm <- DocumentTermMatrix(corpus,
+    control = list(stemming = TRUE, stopwords = TRUE, minWordLength = 3,
+      removeNumbers = TRUE, removePunctuation = TRUE))
R> dim(JSS_dtm)

[1]  348 3273
```

The mean term frequency-inverse document frequency (tf-idf) over documents containing this term is used to select the vocabulary. This measure allows to omit terms which have low frequency as well as those occurring in many documents. We only include terms which have a tf-idf value of at least 0.1 which is a bit less than the median and ensures that the very frequent terms are omitted.

```
R> summary(col_sums(JSS_dtm))

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   1.000   2.000   6.931   5.000 550.000 

R> term_tfidf <-
+    tapply(JSS_dtm$v/row_sums(JSS_dtm)[JSS_dtm$i], JSS_dtm$j, mean) *
+      log2(nDocs(JSS_dtm)/col_sums(JSS_dtm > 0))
R> summary(term_tfidf)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02276 0.08615 0.11400 0.14570 0.16240 1.20600
```

```
R> JSS_dtm <- JSS_dtm[, term_tfidf >= 0.1]
R> JSS_dtm <- JSS_dtm[row_sums(JSS_dtm) > 0,]
R> summary(col_sums(JSS_dtm))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   3.406   4.000  64.000
```

After this pre-processing we have the following document-term matrix with a reduced vocabulary which we can use to fit topic models.

```
R> dim(JSS_dtm)
```

```
[1]  348 2020
```

In the following we fit an LDA model with 30 topics using (1) VEM with α estimated, (2) VEM with α fixed and (3) Gibbs sampling with a burn-in of 1000 iterations and recording every 100th iterations for 1000 iterations. The initial α is set to the default value. By default only the best model with respect to the log-likelihood $\log(p(w|z))$ observed during Gibbs sampling is returned. In addition a CTM is fitted using VEM estimation.

We set the number of topics rather arbitrarily to 30 after investigating the performance with the number of topics varied from 2 to 200 using 10-fold cross-validation. The results indicated that the number of topics has only a small impact on the model fit on the hold-out data. There is only slight indication that the solution with two topics performs best and that the performance deteriorates again if the number of topics is more than 100. For applications a model with only two topics is of little interest because it enables only to group the documents very coarsely. This lack of preference of a model with a reasonable number of topics might be due to the facts that (1) the corpus is rather small containing less than 500 documents and (2) the corpus consists only of text documents on statistical software.

```
R> k <- 30
R> SEED <- 2010
R> jss_TM <- list(
+    VEM = LDA(JSS_dtm, k = k, control = list(seed = SEED)),
+    VEM_fixed = LDA(JSS_dtm, k = k, control = list(estimate.alpha = FALSE,
+      seed = SEED)),
+    Gibbs = LDA(JSS_dtm, k = k, method = "Gibbs", control = list(
+      seed = SEED, burnin = 1000, thin = 100, iter = 1000)),
+    CTM = CTM(JSS_dtm, k = k, control = list(seed = SEED,
+      var = list(tol = 10^-4), em = list(tol = 10^-3))))
```

To compare the fitted models we first investigate the α values of the models fitted with VEM and α estimated and with VEM and α fixed.

```
R> sapply(jss_TM[1:2], slot, "alpha")
```

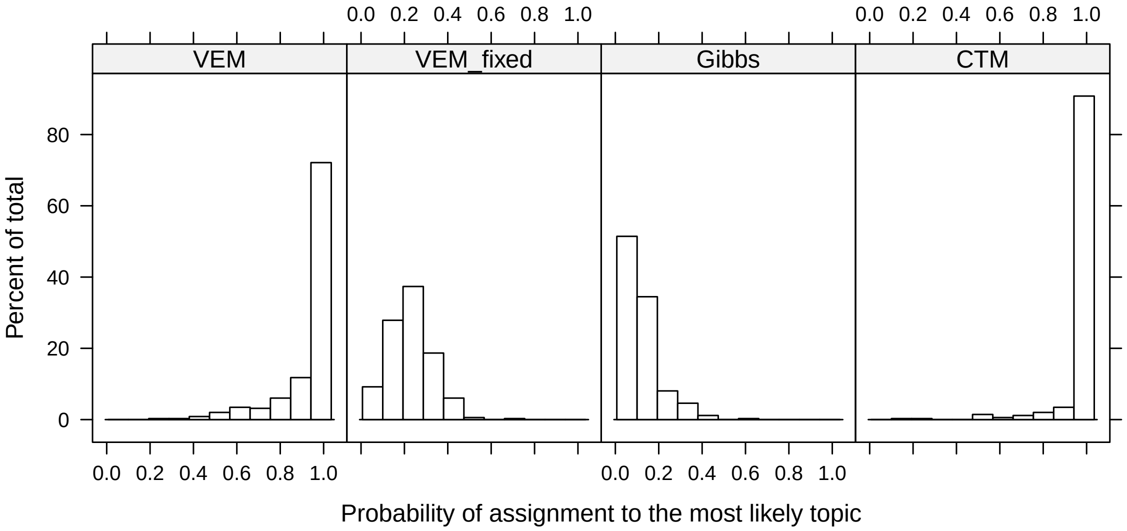


Figure 1: Histogram of the probabilities of assignment to the most likely topic for all documents for the different estimation methods.

```
        VEM   VEM_fixed
0.009669373 1.666666667
```

We see that if α is estimated it is set to a value much smaller than the default. This indicates that in this case the Dirichlet distribution has more mass at the corners and hence, documents consist only of few topics. The influence of α on the estimated topic distributions for documents is illustrated in Figure 1 where the probabilities of the assignment to the most likely topic for all documents are given. The lower α the higher is the percentage of documents which are assigned to one single topic with a high probability. Furthermore, it indicates that the association of documents with only one topic is strongest for the CTM solution.

The entropy measure can also be used to indicate how the topic distributions differ for the four fitting methods. We determine the mean entropy for each fitted model over the documents. The term distribution for each topic as well as the predictive distribution of topics for a document can be obtained with `posterior()`. A list with components `"terms"` for the term distribution over topics and `"topics"` for the topic distributions over documents is returned.

```
R> sapply(jss_TM, function(x) mean(apply(posterior(x)$topics,
+    1, function(z) - sum(z * log(z)))))
```

```
      VEM VEM_fixed     Gibbs       CTM
0.2863427 3.0925014 3.2519352 0.1839297
```

Higher values indicate that the topic distributions are more evenly spread over the topics.

The estimated topics for a document and estimated terms for a topic can be obtained using the convenience functions `topics()` and `terms()`. The most likely topic for each document is obtained by

```
R> Topic <- topics(jss_TM[["VEM"]], 1)
```

The five most frequent terms for each topic are obtained by

```
R> Terms <- terms(jss_TM[["VEM"]], 5)
R> Terms[, 1:5]

     Topic 1       Topic 2  Topic 3      Topic 4     Topic 5
[1,] "constraint"  "robust" "multivari"  "densiti"   "correl"
[2,] "fechnerian"  "genet"  "gene"       "exponenti" "gee"
[3,] "metaanalysi" "intern" "aspect"     "mixtur"    "qls"
[4,] "pattern"     "pilot"  "robust"     "zeroinfl"  "critic"
[5,] "ptak"        "plan"   "microarray" "random"    "hypothes"
```

If any category labelings of the documents were available, these could be used to validate the fitted model. Some JSS papers should have similar content because they appeared in the same special volume. The most likely topic of the papers which appeared in Volume 24 called "Statistical Modeling of Social Networks with **statnet**" (see Handcock, Hunter, Butts, Goodreau, and Morris 2008) is given by

```
R> (topics_v24 <-
+   topics(jss_TM[["VEM"]])[grep("/v24/", JSS_papers[, "identifier"])])

243 244 245 246 247 248 249 250 251
  7   4   7   7  26   7   7  27   7

R> most_frequent_v24 <- which.max(tabulate(topics_v24))
```

The similarity between these papers is indicated by the fact that the majority of the papers have the same topic as their most likely topic. The ten most likely terms for topic 7 are given by

```
R> terms(jss_TM[["VEM"]], 10)[, most_frequent_v24]

 [1] "network"   "ergm"      "popul"     "captur"    "multivari"
 [6] "rcaptur"   "social"    "criterion" "growth"    "ssa"
```

Clearly this topic is related to the general theme of the special issue. This indicates that the fitted topic model was successful at detecting the similarity between papers in the same special issue without using this information.

5. Associated Press data

In the following a subset from the Associated Press data from the First Text Retrieval Conference (TREC-1) 1992 (Harman 1992) is analyzed. The data is available together with the code for fitting LDA models and CTMs on Blei's web page (`http://www.cs.princeton.edu/~blei/lda-c`) and is also contained in package **topicmodels**.

```
R> data("AssociatedPress", package = "topicmodels")
R> dim(AssociatedPress)

[1]  2246 10473
```

It consists of 2246 documents and the vocabulary was already selected to contain only words which occur in more than 5 documents.

```
R> range(col_sums(AssociatedPress))

[1]    6 2073
```

The analysis uses 10-fold cross-validation to evaluate the performance of the models. First, the data set is split into 10 test data sets with the remaining data as training data. Only LDA models are fitted in the following using the different estimation methods. CTM is not fitted because of the high memory demand and longer times required for estimating this model. The complete R code for the simulation as well as the summarization of results is given in the Appendix. Below only the main code parts are presented. First, a random seed is set and indices are randomly drawn to split the data into the 10 folds of training and test data.

```
R> set.seed(0908)
R> folding <- sample(rep(seq_len(10),
+    ceiling(nrow(AssociatedPress)))[seq_len(nrow(AssociatedPress))])
```

With `fold` having values from $1, \ldots, 10$ we estimate the model with the three different variants.

```
R> testing <- which(folding == fold)
R> training <- which(folding != fold)
```

The number of topics are varied from

```
R> topics <- 10 * c(1:5, 10, 20)
```

For VEM with α free, we have:

```
R> train <- LDA(AssociatedPress[training,], k = k,
+    control = list(verbose = 100))
R> test <- LDA(AssociatedPress[testing,], model = train,
+    control = list(estimate.beta = FALSE))
```

For VEM with α fixed:

```
R> train <- LDA(AssociatedPress[training,], k = k,
+    control = list(verbose = 100, estimate.alpha = FALSE))
R> test <- LDA(AssociatedPress[testing,], model = train,
+    control = list(estimate.beta = FALSE))
```

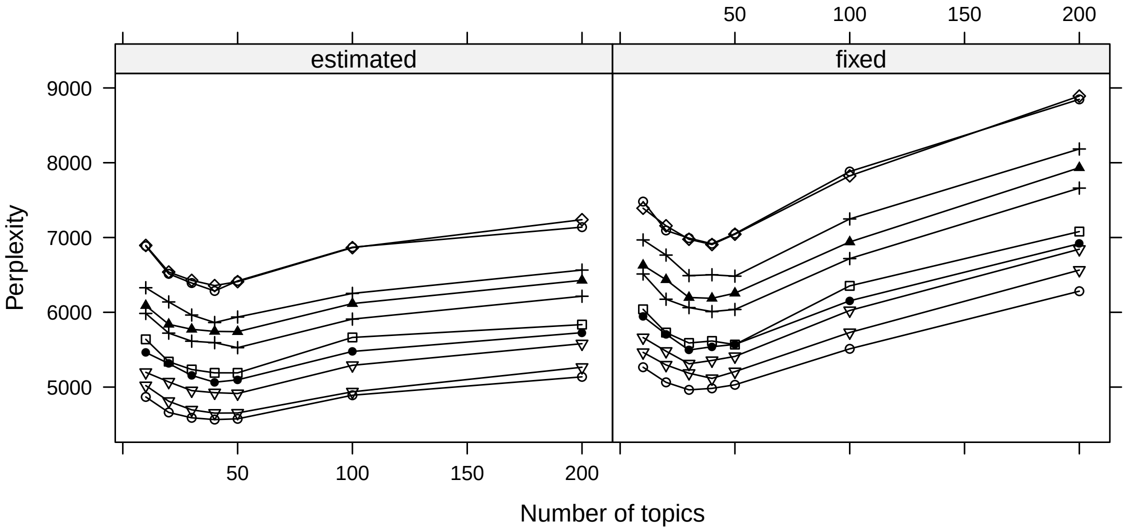


Figure 2: Perplexities of the test data for the models fitted with VEM. Each line corresponds to one of the folds in the 10-fold cross-validation.

For Gibbs sampling we use a burn-in of 1000 followed by 1000 draws with a thinning of 100 and have all draws returned.

```
R> train <- LDA(AssociatedPress[training,], k = k, method = "Gibbs",
+    control = list(burnin = 1000, thin = 100, iter = 1000, best = FALSE))
R> test <- LDA(AssociatedPress[testing,],
+    model = train[[which.max(sapply, train, logLik)]],
+    control = list(estimate.beta = FALSE, burnin = 1000, thin = 100,
+    iter = 1000, best = FALSE))
```

The perplexities of the test data for the models fitted using VEM are given in Figure 2. For both estimation methods about 40 topics are suggested as optimal. The α values estimated by VEM are given in Figure 3 on the left. Obviously these values are much smaller than those used as default with $50/k$. Again, note that small values of α indicate that the topic distribution over documents has most of its weight in the corners. This implies that the documents consist only of a small number of topics. For the model fitted using Gibbs sampling model selection is also performed by determining the perplexities for the test data. Figure 3 on the right suggests that about 20–40 topics are optimal.

We estimate the LDA model with the three different estimation methods with 40 topics to the complete data set. α is initialized as the mean value of the optimal α values in the cross-validation for the models fitted with VEM and α estimated. The R code is given in the Appendix. We compare the topics detected by the VEM algorithm with α estimated and the Gibbs sampling solution after matching the topics. The topics are matched based on the Hellinger distance between the term distributions of the topics using package **clue** (Hornik 2005).

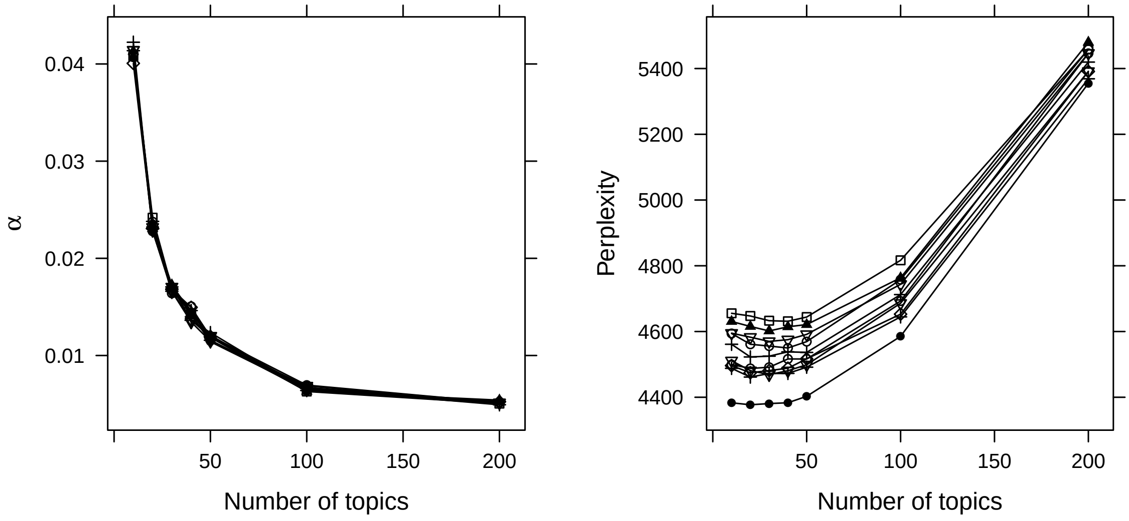


Figure 3: Left: estimated α values for the models fitted using VEM. Right: perplexities of the test data for the models fitted using Gibbs sampling. Each line corresponds to one of the folds in the 10-fold cross-validation.

```
R> dist_models <- distHellinger(posterior(AP[["VEM"]])$terms,
+    posterior(AP[["Gibbs"]])$terms)
R> library("clue")
R> matching <- solve_LSAP(dist_models)
R> dist_models <- dist_models[, matching]
R> d <- mean(diag(dist_models))
```

The best matches with the smallest distance are determined and compared with respect to their eight most likely words.

```
R> best_match <- order(diag(dist_models))
R> terms(AP$VEM, 8)[,best_match[1:4]]
```

```
     Topic 29     Topic 37   Topic 22   Topic 21
[1,] "dukakis"    "company"  "court"    "bill"
[2,] "bush"       "million"  "charges"  "senate"
[3,] "campaign"   "new"      "trial"    "house"
[4,] "democratic" "inc"      "case"     "committee"
[5,] "jackson"    "corp"     "attorney" "rep"
[6,] "i"          "business" "prison"   "legislation"
[7,] "republican" "billion"  "judge"    "sen"
[8,] "president"  "stock"    "drug"     "congress"
```

```
R> terms(AP$Gibbs, 8)[,matching[best_match[1:4]]]
```

```
     Topic 27       Topic 13  Topic 34    Topic 36
[1,] "bush"         "company" "prison"    "bill"
[2,] "dukakis"      "million" "court"     "committee"
[3,] "president"    "inc"     "trial"     "senate"
[4,] "campaign"     "new"     "charges"   "house"
[5,] "jackson"      "stock"   "judge"     "sen"
[6,] "democratic"   "corp"    "attorney"  "rep"
[7,] "presidential" "billion" "convicted" "congress"
[8,] "convention"   "share"   "guilty"    "members"
```

These four topics clearly are on the same subjects and consist of very similar words. However, this clear correspondence between topics is only present for a small subset of the topics. This is also indicated by the image plot of the distances with the matched topics in Figure 4. According to the image we would not expect the worst four matching topics to have much in common. This is also seen by inspecting the eight most important words for each of these topics.

```
R> worst_match <- order(diag(dist_models), decreasing = TRUE)
R> terms(AP$VEM, 8)[, worst_match[1:4]]
```

```
     Topic 14     Topic 13    Topic 26     Topic 27
[1,] "iraq"       "people"    "panama"     "china"
[2,] "government" "estate"    "noriega"    "west"
[3,] "kuwait"     "ireland"   "government" "chinese"
[4,] "iraqi"      "officials" "i"          "east"
[5,] "people"     "three"     "women"      "government"
[6,] "plan"       "ira"       "president"  "years"
[7,] "president"  "army"      "military"   "city"
[8,] "soviet"     "years"     "new"        "german"
```

```
R> terms(AP$Gibbs, 8)[, matching[worst_match[1:4]]]
```

```
     Topic 26  Topic 29    Topic 6      Topic 39
[1,] "cents"   "computer"  "election"   "church"
[2,] "oil"     "company"   "campaign"   "catholic"
[3,] "prices"  "business"  "state"      "pope"
[4,] "futures" "corp"      "republican" "john"
[5,] "cent"    "co"        "vote"       "roman"
[6,] "lower"   "defense"   "percent"    "religious"
[7,] "higher"  "companies" "democratic" "vatican"
[8,] "farmers" "industry"  "candidates" "paul"
```

6. Extending to new fit methods

Package **topicmodels** already provides two different estimation methods for the LDA model and one for the CTM. Users can extend the methods and supply their own fit functions via

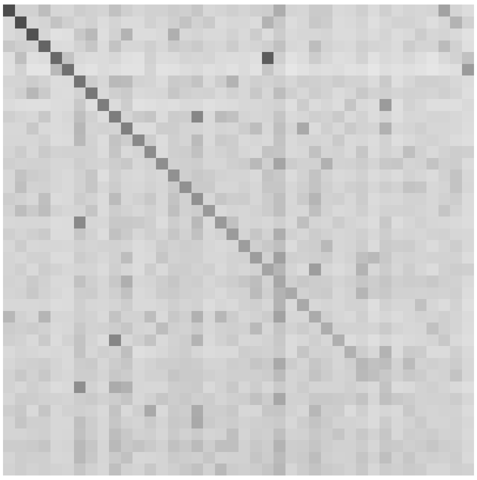


Figure 4: Matched topics of the solution with 40 topics for VEM with free α and with 40 topics for Gibbs sampling.

the `method` argument. In the following we outline how package **rjags** can be used to fit the LDA model using Gibbs sampling with a different implementation. **rjags** is a general purpose Gibbs sampler built on the library **JAGS** (Just another Gibbs sampler; Plummer 2003) and model specification is by a model language not unlike the one for **WinBUGS** (Lunn, Thomas, Best, and Spiegelhalter 2000). The advantage of a general purpose Gibbs sampler is that a wide range of models can be fitted without writing new code for each of the models. This allows to easily try out different models and compare them. However, a drawback is that in general the Gibbs sampler used is less efficient. This is especially a problem in applications where complicated models are fitted and/or large data sets are used. For application of the LDA model the size of the document-term matrix will in general be too large to use a general purpose Gibbs sampler and achieve a reasonable performance. The following code therefore only serves as an illustrative example for extending the package to new methods.

If **rjags** is used for fitting the LDA model, the model needs to be specified in the **JAGS** language. Assuming that the data is given by vectors `D` and `W`. Both vectors have a length equal to the number of words in the document-term matrix. `D` indicates the document and `W` the word from the vocabulary. `n` is the number of documents in the corpus.

```
R> BUGS_MODEL <-
+  "model {
+    for (i in 1:length(W)) {
+      z[i] ~ dcat(theta[D[i],]);
+      W[i] ~ dcat(beta[z[i],]);
+    }
+    for (j in 1:n) {
+      theta[j,1:k] ~ ddirch(alpha);
+    }
+    for (K in 1:k) {
+      beta[K,1:V] ~ ddirch(delta);
+    }
+  }"
```

The following code implements a new method function to fit the LDA model using Gibbs sampling with package **rjags**. In this function for fitting the model the log-likelihood as well as the most probable topic membership of each word in the corpus are not determined and therefore not part of the returned object.

```
R> LDA_rjags.fit <- function(x, k, control = NULL, model = NULL, call, ....)
+  {
+    if (!require("rjags"))
+      stop("\nThis method requires package 'rjags'")
+
+    control <- as(control, "LDA_Gibbscontrol")
+    if (length(control@alpha) == 0)
+      control@alpha <- if (!is.null(model)) model@alpha else 50/k
+
+    DATA <- list(D = rep(x$i, x$v), W = rep(x$j, x$v), n = nrow(x), k = k,
+      V = ncol(x), alpha = rep(control@alpha, k),
+      delta = rep(control@delta, ncol(x)))
+
+    FILE <- file()
+    cat(BUGS_MODEL, file = FILE)
+    model <- jags.model(FILE, data = DATA,
+      inits = list(.RNG.name = "base::Wichmann-Hill",
+      .RNG.seed = control@seed))
+    close(FILE)
+
+    if (control@burnin > 0) update(model, iter = control@burnin)
+    SAMPLES <- coda.samples(model, c("theta", "beta", "z"),
+      thin = control@thin, n.iter = control@iter)[[1]]
+    index_beta <- seq_len(k * ncol(x))
+    index_gamma <- k * ncol(x) + seq_len(nrow(x) * k)
+    obj <- lapply(seq_len(nrow(SAMPLES)), function(i)
+      new("LDA_Gibbs",
+        call = call, Dim = dim(x), k = as.integer(k), control = control,
+        alpha = control@alpha, delta = control@delta,
+        terms = colnames(x), documents = rownames(x),
+        beta = matrix(SAMPLES[i, index_beta], nrow = k),
+        gamma = matrix(SAMPLES[i, index_gamma], ncol = k)))
+    if (nrow(SAMPLES) == 1) obj <- obj[[1]]
+    obj
+  }
```

We apply the new fit function only to a small subset of the Associated Press data and perform only 20 iterations of the Gibbs sampler in order to limit the time needed. The time needed is also compared to the LDA specific implementation of the Gibbs sampler.

```
R> AP_small <- AssociatedPress
R> AP_small <- AP_small[row_sums(AP_small) > 500,]
```

```
R> AP_small <- AP_small[,col_sums(AP_small) > 10,]
R> dim(AP_small)

[1]  18 162

R> system.time({
+    lda <- LDA(AP_small, k = 5, method = "Gibbs",
+      control = list(burnin = 0, iter = 20, seed = 2010))
+  })

   user  system elapsed 
  0.036   0.000   0.036 

R> system.time({
+    lda_rjags <- LDA(AP_small, k = 5, method = LDA_rjags.fit,
+      control = list(burnin = 0, iter = 20, seed = 2010))
+  })

module basemod loaded
module bugs loaded
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 11839

   user  system elapsed 
  5.664   0.048   5.981 

R> terms(lda_rjags, 4)

     Topic 1     Topic 2   Topic 3      Topic 4      Topic 5  
[1,] "gorbachev" "defense" "department" "department" "united" 
[2,] "people"    "soviet"  "leaders"    "people"     "defense"
[3,] "i"         "i"       "new"        "bush"       "new"    
[4,] "held"      "weapons" "president"  "military"   "bush"   
```

As can be seen from this example, adding a new estimation method in package **topicmodels** requires writing a suitable fit function. The fit function takes the document-term matrix, fits the model and returns an object of class "TopicModel" or an extended class thereof. The accessor functions terms() and topics() can then be used to extract the fitted term distributions for the topics as well as the fitted topic distributions for the documents.

7. Summary

Package **topicmodels** provides functionality for fitting the topic models LDA and CTM in R. It builds on and complements functionality for text mining already provided by package

tm. Functionality for constructing a corpus, transforming a corpus into a document-term matrix and selecting the vocabulary is available in **tm**. The basic text mining infrastructure provided by package **tm** is hence extended to allow also fitting of topic models which are seen nowadays as state-of-the-art techniques for analyzing document-term matrices. The advantages of package **topicmodels** are that (1) it gives access within R to the code written by David M. Blei and co-authors, who introduced the LDA model as well as the CTM in their papers, and (2) allows different estimation methods by providing VEM estimation as well Gibbs sampling. Extensibility to other estimation techniques or slightly different model variants is easily possible via the `method` argument.

Packages **Snowball** (Hornik 2009) and **tm** provide stemmers and stop word lists not only for English, but also for other languages. To the authors' knowledge topic models have so far only been used for corpora in English. The availability of all these tools in R hopefully does not only lead to an increased use of these models, but also facilitates to try them out for corpora in other languages as well as in different settings. In addition different modeling strategies for model selection, such as cross-validation, can be easily implemented with a few lines of R code and the results can be analyzed and visualized using already available tools in R.

Due to memory requirements package **topicmodels** will for standard hardware only work for reasonably large corpora with numbers of topics in the hundreds. Gibbs sampling needs less memory than using the VEM algorithm and might therefore be able to fit models when the VEM algorithm fails due to high memory demands. In order to be able to fit topic models to very large data sets distributed algorithms to fit the LDA model were proposed for Gibbs sampling in Newman *et al.* (2009). The proposed Approximate Distributed LDA (AD-LDA) algorithm requires the Gibbs sampling methods available in **topicmodels** to be performed on each of the processors. In addition functionality is needed to repeatedly distribute the data and parameters to the single processors and synchronize the results from the different processors until a termination criterion is met. Algorithms to parallelize the VEM algorithm for fitting LDA models are outlined in Nallapati, Cohen, and Lafferty (2007). In this case the processors are used in the E-step such that each processor calculates only the sufficient statistics for a subset of the data. We intend to look into the potential of leveraging the existing infrastructure for large data sets along the lines proposed in Nallapati *et al.* (2007) and Newman *et al.* (2009).

The package allows us to fit topic models to different corpora which are already available in R using package **tm** or can easily be constructed using tools such as the package **OAIHarvester**. We are also interested in comparing the performance of topic models for clustering documents to other approaches such as using mixtures of von Mises-Fisher distributions to model the term distributions of the documents (Banerjee, Dhillon, Ghosh, and Sra 2005) where the R package **movMF** (Hornik and Grün 2011) is available on CRAN.

Different variants of topic models have been recently proposed. Some models aim at relaxing the assumption of independence of topics which is imposed by LDA such as the CTM, hierarchical topic models (Blei, Griffiths, Jordan, and Tenenbaum 2003a) or Pachinko allocation (Li and McCallum 2006) and hierarchical Pachinko allocation (Mimno, Li, and McCallum 2007). Another possible extension of the LDA model is to include additional information. Using the time information leads to dynamic topic models (Blei and Lafferty 2006) while using the author information of the documents gives the author-topic model (Rosen-Zvi, Chemudugunta, Griffiths, Smyth, and Steyvers 2010). We are interested in extending the package to cover at least a considerable subset of the different proposed topic models. As a starting point we will use Heinrich (2009) and Heinrich and Goesele (2009) who provide a common framework for topic models which only consist of Dirichlet-multinomial mixture "levels". Examples for such topic models are LDA, the author-topic model, Pachinko allocation and hierarchical Pachinko allocation.

Acknowledgments

We would like to thank two anonymous reviewers for their valuable comments which led to several improvements. This research was supported by the the Austrian Science Fund (FWF) under Hertha-Firnberg grant T351-N18 and under Elise-Richter grant V170-N18.

References

Airoldi EM, Blei DM, Fienberg SE, Xing EP (2008). "Mixed Membership Stochastic Blockmodels." *Journal of Machine Learning Research*, **9**, 1981–2014.

Banerjee A, Dhillon IS, Ghosh J, Sra S (2005). "Clustering on the Unit Hypersphere Using von Mises-Fisher Distributions." *Journal of Machine Learning Research*, **6**, 1345–1382.

Blei DM, Griffiths TL, Jordan MI, Tenenbaum JB (2003a). "Hierarchical Topic Models and the Nested Chinese Restaurant Process." In S Thrun, LK Saul, B Schölkopf (eds.), *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Blei DM, Lafferty JD (2006). "Dynamic Topic Models." In *ICML'06: Proceedings of the 23rd International Conference on Machine Learning*, pp. 113–120. ACM Press.

Blei DM, Lafferty JD (2007). "A Correlated Topic Model of Science." *The Annals of Applied Statistics*, **1**(1), 17–35.

Blei DM, Lafferty JD (2009). "Topic Models." In A Srivastava, M Sahami (eds.), *Text Mining: Classification, Clustering, and Applications*. Chapman & Hall/CRC Press.

Blei DM, Ng AY, Jordan MI (2003b). "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, **3**, 993–1022.

Chang J (2010). ***lda**: Collapsed Gibbs Sampling Methods for Topic Models*. R package version 1.2.3, URL `http://CRAN.R-project.org/package=lda`.

Daumé III H (2008). ***HBC**: Hierarchical Bayes Compiler*. Pre-release version 0.7, URL `http://www.cs.utah.edu/~hal/HBC/`.

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990). "Indexing by Latent Semantic Analysis." *Journal of the American Society for Information Science*, **41**(6), 391–407.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data Via the EM-Algorithm." *Journal of the Royal Statistical Society B*, **39**, 1–38.

Feinerer I (2011). ***tm**: Text Mining Package.* R package version 0.5-5., URL http://CRAN.R-project.org/package=tm.

Feinerer I, Hornik K, Meyer D (2008). "Text Mining Infrastructure in R." *Journal of Statistical Software*, **25**(5), 1–54. URL http://www.jstatsoft.org/v25/i05/.

Griffiths TL, Steyvers M (2004). "Finding Scientific Topics." *Proceedings of the National Academy of Sciences of the United States of America*, **101**, 5228–5235.

Hall D, Jurafsky D, Manning CD (2008). "Studying the History of Ideas Using Topic Models." In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A Meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 363–371. ACL.

Handcock MS, Hunter DR, Butts CT, Goodreau SM, Morris M (2008). "**statnet**: Software Tools for the Representation, Visualization, Analysis and Simulation of Network Data." *Journal of Statistical Software*, **24**(1), 1–11. URL http://www.jstatsoft.org/v24/i01/.

Harman D (1992). "Overview of the First Text Retrieval Conference (TREC-1)." In D Harman (ed.), *Proceedings of the First Text Retrieval Conference (TREC-1)*, NIST Special Publication 500-207, pp. 1–20. National Institute of Standards and Technology.

Heinrich G (2009). "A Generic Approach to Topic Models." In WL Buntine, M Grobelnik, D Mladenic, J Shawe-Taylor (eds.), *Machine Learning and Knowledge Discovery in Databases*, volume 5781 of *Lecture Notes in Computer Science*, pp. 517–532. Springer-Verlag, Berlin.

Heinrich G, Goesele M (2009). "Variational Bayes for Generic Topic Models." In B Mertsching, M Hund, Z Aziz (eds.), *KI 2009: Advances in Artificial Intelligence*, volume 5803 of *Lecture Notes in Computer Science*, pp. 161–168. Springer-Verlag, Berlin.

Hoffman MD, Blei DM, Bach F (2010). "Online Learning for Latent Dirichlet Allocation." In J Lafferty, CKI Williams, J Shawe-Taylor, R Zemel, A Culotta (eds.), *Advances in Neural Information Processing Systems 23*, pp. 856–864. MIT Press, Cambridge, MA.

Hofmann T (1999). "Probabilistic Latent Semantic Indexing." In *SIGIR'99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 50–57. ACM Press.

Hornik K (2005). "A CLUE for CLUster Ensembles." *Journal of Statistical Software*, **14**(12), 1–25. URL http://www.jstatsoft.org/v14/i12/.

Hornik K (2009). ***Snowball**: Snowball Stemmers.* R package version 0.0-7, URL http://CRAN.R-project.org/package=Snowball.

Hornik K (2011). ***OAIHarvester**: Harvest Metadata Using OAI-PMH v2.0.* R package version 0.1-3, URL http://CRAN.R-project.org/package=OAIHarvester.

Hornik K, Grün B (2011). ***movMF**: Mixtures of von Mises Fisher Distributions.* R package version 0.0-0, URL http://CRAN.R-project.org/package=movMF.

Hornik K, Meyer D, Buchta C (2011). ***slam**: Sparse Lightweight Arrays and Matrices.* R package version 0.1-21, URL http://CRAN.R-project.org/package=slam.

Li W, McCallum A (2006). "Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations." In *ICML'06: Proceedings of the 23rd International Conference on Machine Learning*, pp. 577–584. ACM Press, New York.

Li Z, Wang C, Xie X, Wang X, Ma WY (2008). "Exploring LDA-Based Document Model for Geographic Information Retrieval." In C Peters, V Jijkoun, T Mandl, H Müller, D Oard, AP nas, V Petras, D Santos (eds.), *Advances in Multilingual and Multimodal Information Retrieval*, volume 5152 of *Lecture Notes in Computer Science*, pp. 842–849. Springer-Verlag, Berlin.

Lunn DJ, Thomas A, Best N, Spiegelhalter D (2000). "WinBUGS—A Bayesian Modelling Framework: Concepts, Structure, and Extensibility." *Statistics and Computing*, **10**(4), 325–337.

McCallum AK (2002). ***MALLET**: Machine Learning for Language Toolkit.* URL http://mallet.cs.umass.edu/.

Microsoft Corporation (2010). ***Infer.NET** User Guide.* Version 2.4 beta 2, URL http://research.microsoft.com/en-us/um/cambridge/projects/infernet/.

Mimno D, Li W, McCallum A (2007). "Mixtures of Hierarchical Topics with Pachinko Allocation." In *ICML'07: Proceedings of the 21st International Conference on Machine Learning*, pp. 633–640. ACM Press.

Mochihashi D (2004a). "A Note on a Variational Bayes Derivation of Full Bayesian Latent Dirichlet Allocation." Unpublished manuscript, URL http://chasen.org/~daiti-m/paper/lda-fullvb.pdf.

Mochihashi D (2004b). ***lda**, a Latent Dirichlet Allocation Package.* MATLAB and C package version 0.1, URL http://chasen.org/~daiti-m/dist/lda/.

Nallapati R, Cohen W, Lafferty J (2007). "Parallelized Variational EM for Latent Dirichlet Allocation: An Experimental Evaluation of Speed and Scalability." In *ICDMW'07: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, pp. 349–354. IEEE Computer Society, Washington, DC.

Newman D, Asuncion A, Smyth P, Welling M (2009). "Distributed Algorithms for Topic Models." *Journal of Machine Learning Research*, **10**, 1801–1828.

Newton MA, Raftery AE (1994). "Approximate Bayesian Inference with the Weighted Likelihood Bootstrap." *Journal of the Royal Statistical Society B*, **56**(1), 3–48.

Nigam K, McCallum AK, Thrun S, Mitchell T (2000). "Text Classification from Labeled and Unlabeled Documents Using EM." *Machine Learning*, **39**(2–3), 103–134.

Phan XH, Nguyen LM, Horiguchi S (2008). "Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-Scale Data Collections." In *Proceedings of the 17th International World Wide Web Conference (WWW 2008)*, pp. 91–100. Beijing, China.

Plummer M (2003). "**JAGS**: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling." In K Hornik, F Leisch, A Zeileis (eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, Austria.* ISSN 1609-395X, URL http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/.

Plummer M (2011). ***rjags**: Bayesian Graphical Models Using MCMC.* R package version 2.2.0-3, URL http://CRAN.R-project.org/package=rjags.

Porteous I, Asuncion A, Newman D, Ihler A, Smyth P, Welling M (2008). "Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation." In *KDD'08: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 569–577. ACM Press.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Rosen-Zvi M, Chemudugunta C, Griffiths T, Smyth P, Steyvers M (2010). "Learning Author-Topic Models from Text Corpora." *ACM Transactions on Information Systems*, **28**(1).

Steyvers M, Griffiths T (2007). "Probabilistic Topic Models." In TK Landauer, DS McNamara, S Dennis, W Kintsch (eds.), *Handbook of Latent Semantic Analysis.* Lawrence Erlbaum Associates.

Steyvers M, Griffiths T (2011). *MATLAB **Topic Modeling** Toolbox 1.4.* URL http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm.

Teh YW, Jordan MI, Beal MJ, Blei DM (2006). "Hierarchical Dirichlet Processes." *Journal of the American Statistical Association*, **101**(476), 1566–1581.

Temple Lang D (2010). ***XML**: Tools for Parsing and Generating XML Within R and S-PLUS.* R package version 3.2-0, URL http://CRAN.R-project.org/package=XML.

Wainwright MJ, Jordan MI (2008). "Graphical Models, Exponential Families, and Variational Inference." *Foundations and Trends in Machine Learning*, **1**(1–2), 1–305.

Wallach HM, Murray I, Salakhutdinov R, Mimno D (2009). "Evaluation Methods for Topic Models." In *ICML'09: Proceedings of the 26th International Conference on Machine Learning*, pp. 1105–1112. ACM Press.

Wei X, Croft WB (2006). "LDA-Based Document Models for Ad-Hoc Retrieval." In *SIGIR'06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 178–185. ACM Press, New York.

A. R code for the Associated Press analysis

In the following the R code is provided for the analysis of the Associated Press data set. Due to the long duration of the simulation as well as the fitting of the LDA model to the complete data set the results are saved and loaded for the analysis in the paper.

A.1. Simulation using 10-fold cross-validation

```
set.seed(0908)
topics <- 10 * c(1:5, 10, 20)
SEED <- 20080809
library("topicmodels")
data("AssociatedPress", package = "topicmodels")
D <- nrow(AssociatedPress)
folding <-
  sample(rep(seq_len(10), ceiling(D))[seq_len(D)])
for (k in topics) {
  for (chain in seq_len(10)) {
    FILE <- paste("VEM_", k, "_", chain, ".rda", sep = "")
    training <- LDA(AssociatedPress[folding != chain,], k = k,
      control = list(seed = SEED))
    testing <- LDA(AssociatedPress[folding == chain,], model = training,
      control = list(estimate.beta = FALSE, seed = SEED))
    save(training, testing, file = file.path("results", FILE))
    FILE <- paste("VEM_fixed_", k, "_", chain, ".rda", sep = "")
    training <- LDA(AssociatedPress[folding != chain,], k = k,
      control = list(seed = SEED, estimate.alpha = FALSE))
    testing <- LDA(AssociatedPress[folding == chain,], model = training,
      control = list(estimate.beta = FALSE, seed = SEED))
    save(training, testing, file = file.path("results", FILE))
    FILE <- paste("Gibbs_", k, "_", chain, ".rda", sep = "")
    training <- LDA(AssociatedPress[folding != chain,], k = k,
      control = list(seed = SEED, burnin = 1000, thin = 100,
      iter = 1000, best = FALSE), method = "Gibbs")
    best_training <- training@fitted[[which.max(logLik(training))]]
    testing <- LDA(AssociatedPress[folding == chain,],
      model = best_training, control = list(estimate.beta = FALSE,
        seed = SEED, burnin = 1000, thin = 100, iter = 1000, best = FALSE))
    save(training, testing, file = file.path("results", FILE))
  }
}
```

A.2. Summarizing the cross-validation simulation results

```
set.seed(0908)
```

```
topics <- 10 * c(1:5, 10, 20)
library("topicmodels")
data("AssociatedPress", package = "topicmodels")
D <- nrow(AssociatedPress)
folding <-
  sample(rep(seq_len(10), ceiling(D))[seq_len(D)])
AP_test <- AP_alpha <- list()
for (method in c("VEM", "VEM_fixed", "Gibbs")) {
  AP_alpha[[method]] <- AP_test[[method]] <- matrix(NA,
    nrow = length(topics), ncol = 10, dimnames = list(topics, seq_len(10)))
  for (fold in seq_len(10)) {
    for (i in seq_along(topics)) {
      T <- topics[i]
      FILE <- paste(method, "_", T, "_", fold, ".rda", sep = "")
      load(file.path("results", FILE))
      AP_alpha[[method]][paste(T),fold] <-
        if (is(training, "Gibbs_list")) training@fitted[[1]]@alpha
        else training@alpha
      AP_test[[method]][paste(T),fold] <- perplexity(testing,
        AssociatedPress[folding == fold,], use_theta = FALSE)
    }
  }
}
save(AP_alpha, AP_test, file = "AP.rda")
```

A.3. Fitting the LDA model to the complete data set using 40 topics

```
library("topicmodels")
data("AssociatedPress", package = "topicmodels")
topics <- 10 * c(1:5, 10, 20)
k <- 40
load("AP.rda")
alpha <- mean(AP_alpha[["VEM"]][which(topics == k),])
rm(AP_alpha, AP_test)
SEED <- 20080806
AP <- list(
  VEM = LDA(AssociatedPress, k = k,
    control = list(alpha = alpha, seed = SEED)),
  VEM_fixed = LDA(AssociatedPress, k = k,
    control = list(alpha = alpha, estimate.alpha = FALSE, seed = SEED)),
  Gibbs = LDA(AssociatedPress, k = k, method = "Gibbs",
    control = list(alpha = alpha, seed = SEED, burnin = 1000,
    thin = 100, iter = 1000)))
save(AP, file = "AP-40.rda")
```

Affiliation:

Bettina Grün
Institut für Angewandte Statistik / IFAS
Johannes Kepler Universität Linz
Altenbergerstraße 69
4040 Linz, Austria
E-mail: `Bettina.Gruen@jku.at`
URL: `http://ifas.jku.at/gruen/`

Kurt Hornik
Institute for Statistics and Mathematics
WU Wirtschaftsuniversität Wien
Augasse 2–6
1090 Wien, Austria
E-mail: `Kurt.Hornik@R-project.org`
URL: `http://statmath.wu.ac.at/~hornik/`

Journal of Statistical Software `http://www.jstatsoft.org/`
published by the American Statistical Association `http://www.amstat.org/`
Volume 40, Issue 13 *Submitted:* 2010-09-23
May 2011 *Accepted:* 2011-04-26